

## מגמת הנדסת אלקטרוניקה ומחשבים

תכנית לימודים במקצוע

**תכנות בשפת Python**

**לפי תוכנית לימודים של תיכון וגם של מכללה.  
ספר לימוד למורה ולתלמיד : תיאוריה, דוגמאות ותרגילים.**

כתב: מר מרק טסליצקי

הגהה: ד"ר אודי לביא.

גרסה 3.

יוני 2022



## תוכן עניינים:

2	..... תוכן עניינים:
8	..... פרק א. "הדרך מתחילה מצעד ראשון" (קונפוזיוס).
8	..... הקדמה:
8	..... הזן של פייתון:
9	..... סימנים מיוחדים בספר:
10	..... פרק ב'. מושגים בסיסיים.
10	..... מבוא לשפת פייתון.
10	..... "משתנים דינמיים".
10	..... תמיכה מודולארית טובה.
11	..... תמיכה מקורית ב- Unicode במחרוזות.
11	..... תמיכה בתכנות מונחה עצמים.
11	..... "איסוף אשפה" אוטומטי, ללא דליפות זיכרון.
11	..... שילוב עם C / ++ C אם תכנות פייתון אינך מספיקות.
11	..... תחביר ברור ותמציתי התורם להצגת קוד ברורה.
11	..... מספר עצום של מודולים,
12	..... "חוצה פלטפורמות".
12	..... התקנת הסביבה.
12	..... סביבת interpreter.
16	..... כדי לקבל עזרה נשתמש בפקודה help().
20	..... מערכת PyCharm.
27	..... הוספת מודולים בעזרת מערכת PIP.
35	..... פונקציה print – הדפסת נתונים על המסך (או בשלב מתקדם לתוך הקובץ).
36	..... פונקציה print() למתקדמים:
38	..... שמירת נתונים לתוך הקובץ:
39	..... משתנים, מספרים ופעולות.
39	..... צורת ההשמה המיוחדת שיש בפייתון:
40	..... ניתן להגדיר מספרים בבסיס שונה מבסיס 10.
41	..... פעולת החלפת נתונים:
42	..... משתנים מסוג מחרוזת – String.
42	..... פעולות עם מחרוזות:

44	פונקציות נוספות עם תווים ומחרוזות:
45	משתנים בוליאניים.
46	פעולות לוגיות עם משתנים בוליאניים.
47	אופרטורים מיוחדים "any", "all":
49	פעולות לוגיות מורכבות.
50	משתנים ומספרים שלמים (int).
51	פעולות על סיביות בעזרת מספרים (משתנים) שלמים.
51	הצגת מספר בבסיסים שונים:
52	פונקציה <code>abs()</code>
52	פונקציה <code>round()</code>
53	משתנים מרוכבים ופעולות עם מספרים מרוכבים, (complex):
54	פונקציות מיוחדות של קבוצת <code>fractions</code> .
56	פונקציית קליטת נתונים <code>input</code> .
56	הפיכת מחרוזת למספר.
58	שינוי סוגי נתונים.
60	גנרטור של מספר אקראי:
64	קריאה ושימוש במודולים.
64	כללים לגבי שימוש במודולים:
68	מודול של מספרים מרוכבים – מודול <code>cmath</code> .
69	תרגול בנושא פעולות בסיסיות:
69	תרגול בנושא אופרטורים בסיסיים.
82	תרגילים בנושא טבלאות מעקב.
99	תרגילים בנושא כתיבת קוד.
102	פרק ג. ביצוע מותנה.
102	מבוא לאלגוריתמים.
102	הגדרת של אירועים:
103	תוכנת סרטוט תרשמי זרימה.
103	דוגמאות של תרשימי זרימה:
106	אופרטורים של התניה.
106	אופרטור <code>if</code> .
107	אופרטור <code>if else</code> .
111	משפט "elif".
113	אופרטור שורה <code>if-else</code> .
115	משפט <code>switch-case</code> :
118	טיפול בשגיאות.
119	חריגות מסוג <code>ValueError</code> ו- <code>TypeError</code> .

120.....	חריגה חילוק ב-0.
120.....	חריגה של פונקציות מתמטיות:
120.....	טיפול בחריגות: try except (דומה למנגנון try and catch בשפת C#).
121.....	אם ישנם מספר שגיאות, עדיף לטפל בכל סוג שגיאה בנפרד:
122.....	פקודות נוספות: " else ", " finally ".
125.....	תרגול בנושא ביצוע מותנה:
125.....	תרגול בנושא תרשימי זרימה.
131.....	תרגול בנושא טבלאות מעקב.
163.....	הפיכת קוד מצורה אחד לצורה אחרת.
186.....	תרגול בנושא כתיבת קוד.
192.....	פרק ד. ביצוע חוזר.
192.....	לולאת while.
192.....	שני חריגות ללולאת while:
195.....	דוגמה – בניית סדרת פיבונצ'י.
197.....	לולאה Do While.
197.....	לולאת for.
198.....	אופרטורים break ו- continue.
198.....	אופרטור else בתוך לולאה:
200.....	פונקציית range().
202.....	פונקציה enumerate():
202.....	לולאה בתוך לולאה – for בתוך for.
208.....	תרגול בנושא ביצוע חוזר:
208.....	תרגול בנושא טבלאות מעקב.
242.....	תרגול בנושא הפיכת קוד מוצרה אחד לצורה אחרת.
276.....	תרגול בנושא כתיבת קוד.
281.....	תרגול בנושא לולאה בתוך לולאה – טבלאות מעקב.
295.....	תרגול בנושא לולאה בתוך לולאה – כתיבת קוד.
298.....	פרק ה'. פונקציות.
299.....	הגדרת הפונקציות def.
301.....	פונקציות הופכות תוכנה למובנת ומסודרת.
301.....	נכתוב תוכנית פעם ללא פונקציות ופעם עם פונקציות:
303.....	האם סדר של הגדרות של פונקציות חשוב?
303.....	מושג "*args" ומושג "**kwargs".
306.....	משתנים לוקליים וגלובליים:
309.....	אין בעיה אם אנו נפנה למשתנים גלובליים מפונקציה ראשית.
312.....	החזרת פרמטרים מהפונקציות.

315.....	החזרת מספר ערכים מהפונקציה.....
316.....	פרמטרים וארגומנטים של פונקציות (arguments, parameters).....
318.....	כמות ארגומנטים משתנה.....
319.....	שימוש במושג yield במקום מושג return.....
320.....	גנרטורים של פייתון.....
321.....	תרגול בנושא פונקציות:.....
321.....	תרגול בנושא טבלאות מעקב:.....
340.....	תרגול בנושא כתיבת קוד.....
345.....	פרק ו'. רשימות. Lists.....
345.....	הגדרת רשימות.....
346.....	פעולות בסיסיות עם רשימות.....
347.....	שגיאות בעבודה עם רשימות:.....
347.....	פונקציות נוספות בנושא רשימות.....
352.....	דוגמאות בנושא רשימות (רמה מתקדמת):.....
361.....	תרגול בנושא רשימות:.....
361.....	תרגול בנושא טבלאות מעקב.....
385.....	תרגול בנושא כתיבת קוד.....
394.....	פרק ז'. מערך Array.....
394.....	מה זה מערך ומה הבדל בין רשימה למערך.....
397.....	פונקציות מיוחדות של מערכים:.....
402.....	תרגול בנושא מערכים.....
402.....	תרגול בנושא טבלאות מעקב.....
411.....	תרגול בנושא כתיבת קוד.....
417.....	פרק ח'. רשימות דו-ממדיות (מטריצות).....
417.....	מה זה מטריצה?.....
419.....	דוגמאות:.....
422.....	ספרייה מיוחדת לטיפול במטריצות numpy.....
428.....	תרגול בנושא רשימות דו-ממדיות.....
428.....	תרגול בנושא טבלאות מעקב:.....
452.....	תרגול בנושא כתיבת קוד.....
461.....	פרק ט'. פונקציות מיוחדות של מחרוזות.....
462.....	רשימת פונקציות לטיפול במחרוזות.....
463.....	פונקציות לטיפול במחרוזות עם הסברים ו דוגמאות.....
467.....	תרגול במחרוזות:.....
467.....	תרגול בנושא טבלאות מעקב.....
491.....	תרגול בנושא כתיבת קוד.....

498.....	פרק י'. מבנה נתונים מסוג Tuple - צירוף.
498.....	מה זה Tuple - צירוף.
498.....	הגדרת הצירוף.
499.....	גישה לאיברים של צירוף.
502.....	שינוי והוספת איברים לצירוף.
504.....	ניתן להגדיר רשימות בתוך צירוף.
505.....	מונחים "ארזיה" ו"פריקה". (Packing and Unpacking).
505.....	כדי להימנע משינוי רשימות המקוריות, לא חייבים להשתמש בצירוף.
507.....	תרגול בנושא צירופים.
507.....	תרגול בנושא טבלאות מעקב.
514.....	תרגול בנושא כתיבת קוד.
517.....	פרק י"א'. מילון.
518.....	אופרטורים ששייכים לקבוצה של מילון הם:
519.....	אופרטורים נוספים של מחלקת מילון.
521.....	תרגול בנושא מילונים:
521.....	תרגול בנושא טבלאות מעקב.
531.....	תרגול בנושא כתיבת קוד.
536.....	פרק י"ב'. סטים. Sets
536.....	מה זה סט?
537.....	אופרטורים בין סטים הן:
539.....	ניתן לבצע את אותם הפעולות בעזרת פונקציות מיוחדות:
540.....	חוץ מפונקציות שראינו ישנם עוד מספר פונקציות בנוגע לסטים:
540.....	פונקציות נוספות בנוגע לסטים.
541.....	דוגמה לעבודה עם סטים:
544.....	תרגילים בנושא סטים:
544.....	תרגול בנושא טבלאות מעקב.
557.....	תרגול בנושא כתיבת קוד.
561.....	פרק י"ג'. קבצים.
561.....	אחסון נתונים בתוך קבצים.
561.....	פונקציה open() - פתיחת קובץ.
562.....	קריאת תוכן הקובץ - פונקציה read().
562.....	קריאת שורה מהקובץ - פונקציה readline().
563.....	קריאת קובץ שורה לאחר שורה - פונקציה readlines().
564.....	כתיבה לקובץ - פונקציה write().
564.....	סגירת קובץ - פונקציות close(), closed().
565.....	תרגול בנושא קבצים.

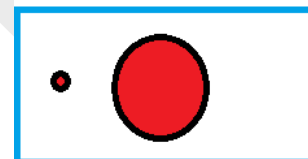
- 565.....תרגול בנושא טבלאות מעקב.
- 577.....תרגול בנושא כתיבת קוד.
- 584.....בנוס:
- 588.....תנסו לבדוק את עצמכם ולענות על השאלון הקצר:
- 589.....דף נוסחאות.

טסליצקי מרק

## פרק א. "הדרך מתחילה מצעד ראשון" (קונפוציוס).

### הקדמה:

מספר מילים על שפת פייתון. כל השפות שהיו עד להגעת שפת פייתון, הן שפות עם חוקים מאוד ברורים וישרים – שחור זה שחור ולבן זה לבן. שפת פייתון היא שפה גמישה מאוד – אפילו אין צורך להגדיר משתנים כמו בשפת C, כמעט הכול מותר כאן. ומכאן הפילוסופיה של פייתון. שפה חינוכית עם מודולים שכל מי שמפתח אותם שם בחינם בענן, דבר חסר תקדים בעולם התכנות. וזה הזן של הפייתון. ננסה להבין את הפילוסופיה הזאת יחד ללא חשש, כי הכול אפשרי. העולם הזה הוא אין סופי. נניח שמה שנמצא בתוך המעגל – זה מה שאנחנו יודעים. מה שבחוץ, זה מה שלא יודעים. היקף המעגל זה מה שאנחנו מבינים שאנחנו לא יודעים. המסקנה המתבקשת כאן היא שככל שיוודעים יותר, מבינים כמה עוד דברים מעניינים יש לנו ללמוד. שיהיה לכם בהצלחה!!!!!!!!!!!!!!!!!!!!



### הזן של פייתון:

**במידה ואתם רושמים בתוך אינטרפרטר: >>>import this , אתם מקבלים קיצור של כל הפילוסופיה של שפת פייתון:**

>>> import this

The Zen of Python, by Tim Peters

Beautiful is better than ugly.
Explicit is better than implicit.
Simple is better than complex.
Complex is better than complicated.
Flat is better than nested.
Sparse is better than dense.
Readability counts.
Special cases aren't special enough to break the rules.
Although practicality beats purity.
Errors should never pass silently.
Unless explicitly silenced.
In the face of ambiguity, refuse the temptation to guess.
There should be one-- and preferably only one --obvious way to do it.
Although that way may not be obvious at first unless you are Dutch.
Now is better than never.
Although never is often better than *right* now.
If the implementation is hard to explain, it's a bad idea.
If the implementation is easy to explain, it may be a good idea.
Namespaces are one honking great idea -- let us do more of those!



## סימנים מיוחדים בספר:

קוד שכתוב בצורת Interpreter יחד עם פלט מיוצג כך:

```
>>> time.time()
1376047104.056417
>>> random.random()
0.9874550833306869
```

קוד שמוגדר בתור Script (סקריפט) מיוצג ללא סימן >>>

קלט ופלט

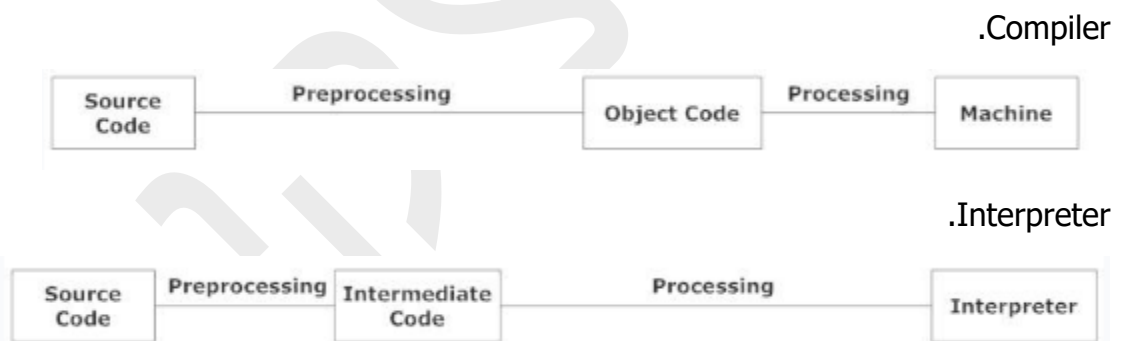
הסברים:

תרגילים (שלא נמצאים בסוף של כל פרק):

## פרק ב'. מושגים בסיסיים. מבוא לשפת פייתון.

פייתון היא שפת תכנות "המתפרשת" (Interpreter). מצד אחד זה יכול לפשט משמעותית את ניפוי הבאגים של התוכנית, ומצד שני זה גורם למהירות ביצוע נמוכה יחסית.

Interpreter	Compiler
מתרגמת פקודה לאחר פקודה לשפת מכונה.	בודק תוכנה כולה ומתרגמת אותה לשפת מכונה.
לוקח פחות זמן לנתח את קוד המקור אך זמן הביצוע הכולל הוא איטי יותר.	לוקח זמן רב לנתח את קוד המקור אך זמן הביצוע הכולל מהיר יחסית.
לא נוצר קוד אובייקט ביניים, ולכן ניצול הזיכרון יעיל יותר.	מייצר קוד אובייקט ביניים שדורש קישור נוסף, ולכן דורש יותר זיכרון.
ממשיך בתרגום התוכנית עד למציאת השגיאה הראשונה, ובמקרה זה היא נפסקת. מכאן נובע איתור באגים קל יותר.	הוא יוצר את הודעת השגיאה רק לאחר סריקת התוכנית כולה. מכאן נובע איתור באגים (Debug) קשה יחסית.



### "משתנים דינמיים".

בפייתון אין צורך להכריז מראש על סוג משתנה, וזה מאוד נוח במהלך הפיתוח.

```
>>>a = 1; b = 2; print(a, b)
```

כפי שרואים אין צורך להגדיר סוג המשתנים כמו בשפת C.

תמיכה מודולארית טובה.

ניתן לכתוב בקלות את המודול שלך ולהשתמש בו בתוכניות אחרות.

הסבר:  
כדי לבצע הכנסת מודולים חיצוניים משתמשים במילה השמורה "import". ניתן לקשר מספר מודולים באותה שורה.

```
>>> import time, random
>>> time.time()
1376047104.056417
>>> random.random()
0.9874550833306869
```

תמיכה מקורית ב- Unicode במחרוזות.  
ב- Python, אין צורך לכתוב הכל באנגלית: עקב תמיכה בתקן Unicode שתומך בשפות רבות, ניתן לרשום בכל שפה שתומך בה Unicode.

```
>>> c = "זה טקסט"
```

תמיכה בתכנות מונחה עצמים.  
לעומת שפות אחרות, בפייתון יש תמיכה מובנית ברמת המקור, בעצמים ובתכנות מונחה עצמים.



"איסוף אשפה" אוטומטי, ללא דליפות זיכרון.  
נדבר על זה מאוחר יותר.

שילוב עם C / ++ C אם תכונות פייתון אינן מספיקות.  
תחביר ברור ותמציתי התורם להצגת קוד ברורה.  
מערכת פונקציות נוחה מאפשרת לך ליצור קוד מונגש, שיהיה קל לאדם אחר להבין במידת הצורך.  
ניתן גם ללמוד לקרוא תוכניות ומודולים שנכתבו על ידי אנשים אחרים.

```
>>>a = 1; b = 2; print(a, b)
```

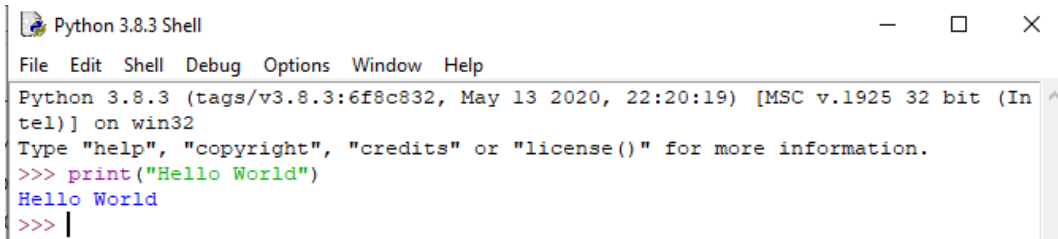
מספר עצום של מודולים,

גם כלולים בחבילה הסטנדרטית של Python 3, ושל צד שלישי. במקרים מסוימים, כדי לכתוב תוכנית אתה רק צריך למצוא את המודולים הנכונים ולשלב אותם נכון. כך תוכלו לחשוב על יצירת תוכנית ברמה גבוהה יותר, לעבוד עם אלמנטים מוכנים המבצעים פעולות שונות.

"חוצה פלטפורמות".

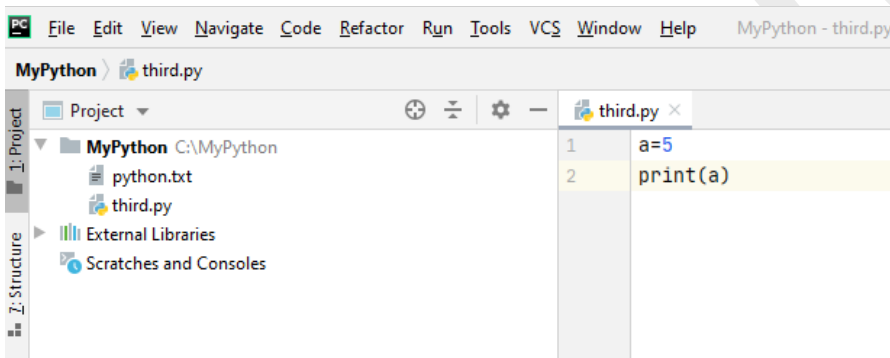
תוכנית שנכתבה בפיתוח תפעל בדיוק אותו דבר ללא קשר למערכת ההפעלה בה היא פועלת. ההבדלים מתעוררים רק במקרים נדירים, וקל לחזות אותם בגלל זמינות התיעוד המפורט.

## בחוברת הזאת ישנן דוגמאות שבניתו בעזרת 2 תוכנות: חלק בנוי בעזרת "מתרגם" – interpreter



```
Python 3.8.3 Shell
File Edit Shell Debug Options Window Help
Python 3.8.3 (tags/v3.8.3:6f8c832, May 13 2020, 22:20:19) [MSC v.1925 32 bit (Intel)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> print("Hello World")
Hello World
>>> |
```

## וחלק בנוי בעזרת תוכנה שקודם כותבים את הקוד ואז מריצים אותו (בלי להפעיל שורה אחרי שורה).



```
MyPython - third.py
File Edit View Navigate Code Refactor Run Tools VCS Window Help
MyPython > third.py
Project
MyPython C:\MyPython
python.txt
third.py
External Libraries
Scratches and Consoles
third.py
1 a=5
2 print(a)
```

## קל לזהות הבדל ביניהם – הראשון בא עם סימן ">>>" והשני בלי.

התקנת הסביבה.

ישנן 2 אפשרויות עיקריות לעבודה עם פיתוח:

סביבת interpreter וסביבת compiler (במקרה של קומפיילר קובץ קוד המקור נקראת "סקריפט" (Script).

סביבת interpreter

קעת נבצע התקנה מהאתר הרשמי של פיתוח: <https://www.python.org/>

Python PSF Docs PyPI Jobs Community

python™ [Donate](#)  [GO](#) [Socialize](#)

[About](#) [Downloads](#) [Documentation](#) [Community](#) [Success Stories](#) [News](#) [Events](#)

```
# Python 3: Fibonacci series up to n
>>> def fib(n):
>>>     a, b = 0, 1
>>>     while a < n:
>>>         print(a, end=' ')
>>>         a, b = b, a+b
>>>     print()
>>> fib(1000): 384
0 1 1 2 3 5 8 13 21 34 55 89 144 233 377
610 987
```

**Functions Defined** [Expect](#)

The core of extensible programming is defining functions. Python allows mandatory and optional arguments, keyword arguments, and even arbitrary argument lists. [More about defining functions in Python 3](#)

[1](#) [2](#) [3](#) [4](#) [5](#)

Python is a programming language that lets you work quickly and integrate systems more effectively. [>>> Learn More](#)

[Downloads](#) [Documentation](#) [Community](#) [Success Stories](#) [News](#)

- All releases
- Source code
- Windows
- Mac OS X
- Other Platforms
- License
- Alternative Implementations

**Download for Windows**

Python 3.8.3

**Note that Python 3.5+ cannot be used on Windows XP or earlier.**

Not the OS you are looking for? Python can be used on many operating systems and environments. [View the full list of downloads.](#)



## Install Python 3.8.3 (32-bit)

Select Install Now to install Python with default settings, or choose Customize to enable or disable features.



**Install Now**

C:\Users\marka\AppData\Local\Programs\Python\Python38-32

Includes IDLE, pip and documentation  
Creates shortcuts and file associations



**Customize installation**

Choose location and features

Install launcher for all users (recommended)

Add Python 3.8 to PATH

Cancel



## Setup Progress

Installing:

Python 3.8.3 Standard Library (32-bit)



Cancel



# Setup was successful

Special thanks to Mark Hammond, without whose years of freely shared Windows expertise, Python for Windows would still be Python for DOS.

New to Python? Start with the [online tutorial](#) and [documentation](#).

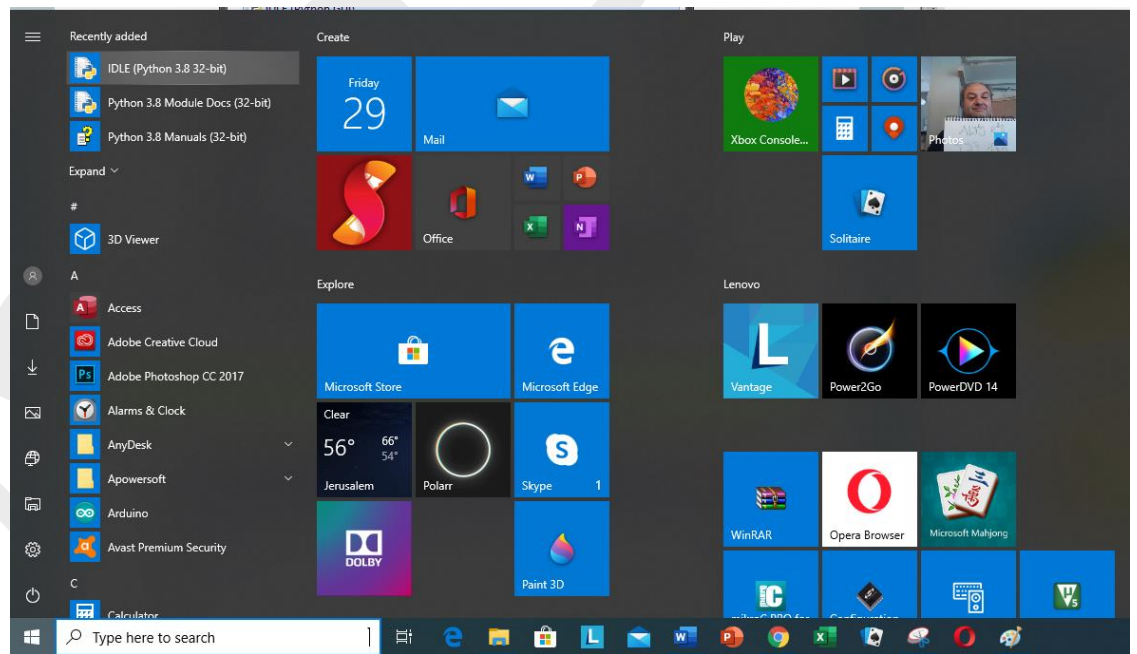
See [what's new](#) in this release.



### Disable path length limit

Changes your machine configuration to allow programs, including Python, to bypass the 260 character "MAX\_PATH" limitation.

Close

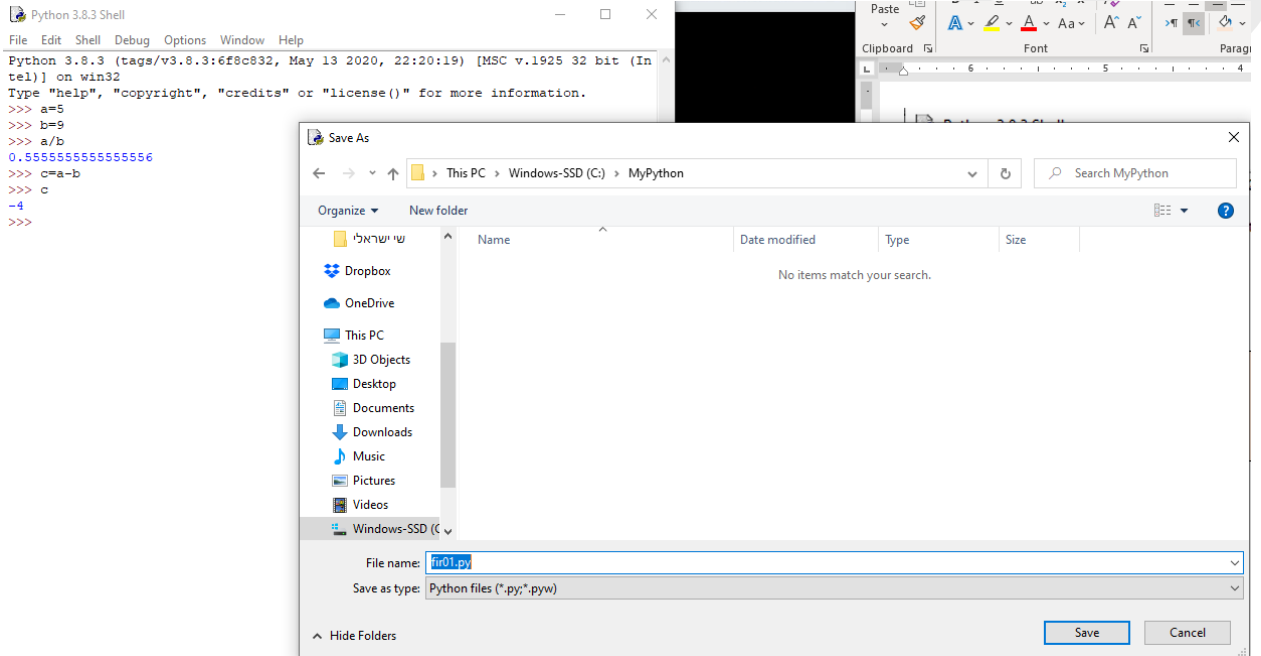


File Edit Shell Debug Options Window Help

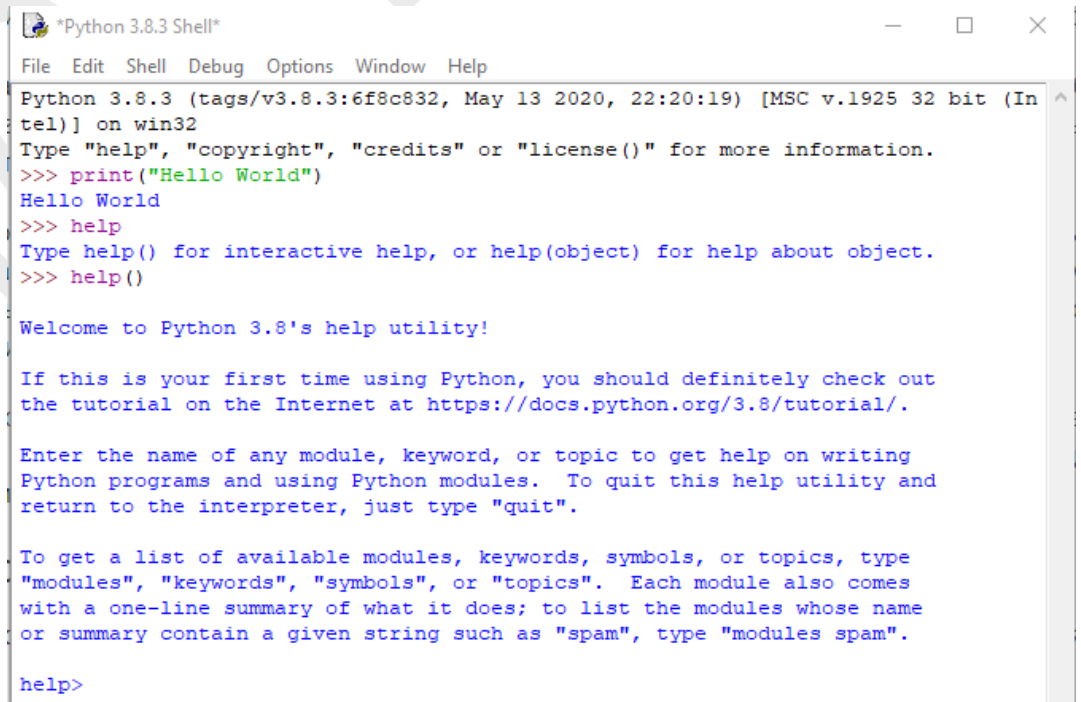
```
Python 3.8.3 (tags/v3.8.3:6f8c832, May 13 2020, 22:20:19) [MSC v.1925 32 bit (Intel)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> |
```

```
>>> 2+3
5
>>> 5-9
-4
```

ניתן לשמור את הקובץ עם התוכנה שכתבנו, כדי לפתוח אותו מאוחר יותר ולהמשיך לעבוד עם הקובץ:



כדי לקבל עזרה נשתמש בפקודה `.help()`.





כאשר ניכנס לHELP – נוכל לבחור תת-תפריט של: modules, keywords, topics, symbols.

```
help> topics

Here is a list of available topics.  Enter any topic name to get more help.

ASSERTION          DELETION           LOOPING            SHIFTING
ASSIGNMENT         DICTIONARIES      MAPPINGMETHODS   SLICINGS
ATTRIBUTEMETHODS  DICTIONARYLITERALS  MAPPINGS         SPECIALATTRIBUTES
ATTRIBUTES         DYNAMICFEATURES   METHODS          SPECIALIDENTIFIERS
AUGMENTEDASSIGNMENT ELLIPSIS          MODULES          SPECIALMETHODS
BASICMETHODS       EXCEPTIONS         NAMESPACES      STRINGMETHODS
BINARY            EXECUTION          NONE            STRINGS
BITWISE           EXPRESSIONS       NUMBERMETHODS   SUBSCRIPTS
BOOLEAN           FLOAT              NUMBERS         TRACEBACKS
CALLABLEMETHODS   FORMATTING        OBJECTS         TRUTHVALUE
CALLS             FRAMEOBJECTS      OPERATORS       TUPLELITERALS
CLASSES           FRAMES            PACKAGES        TUPLES
CODEOBJECTS       FUNCTIONS         POWER           TYPEOBJECTS
COMPARISON        IDENTIFIERS       PRECEDENCE     TYPES
COMPLEX           IMPORTING         PRIVATENAMES    UNARY
CONDITIONAL       INTEGER           RETURNING       UNICODE
CONTEXTMANAGERS   LISTLITERALS     SCOPING
CONVERSIONS       LISTS             SEQUENCEMETHODS
DEBUGGING         LITERALS         SEQUENCES
```

```
help> |
```

```
help> symbols
```

Here is a list of the punctuation symbols which Python assigns special meaning to. Enter any symbol to get more help.

```
!=          +          <=
"           +=         <>
"""        ,          ==         b"
%          -          >         b'
%=         -=         >=        f"
&         .          >>       f'
&=        ...        >>=      j
'         /          @         r"
'''        //         J         r'
(         // =       [         u"
)         /=        \         u'
*         :          ]         |
**        <         ^         |=
**=       <<        ^=        ~
*=        <<=      -
```

```
help> |
```

```

help> modules

Please wait a moment while I gather a list of all available modules...

__future__      asyncio        history        scrolledlist
__main__        asyncore      hmac           search
_abc            atexit       html          searchbase
_ast            audiopop     http          searchengine
_asyncio       autocomplete hyperparser    secrets
_bisect        autocomplete_w idle           select
_blake2        autoexpand   idle_test     selectors
_bootlocale    base64       idllib        setuptools
_bz2           bdb          imaplib       shelve
_codecs        binascii     imghdr        shlex
_codecs_cn     binhex      imp           shutil
_codecs_hk     bisect      importlib     sidebar
_codecs_iso2022 browser      inspect       signal
_codecs_jp     builtins    io            site
_codecs_kr     bz2         iomenu        smtpd
_codecs_tw     cProfile    ipaddress     smtplib
_collections   calendar    itertools     sndhdr
_collections_abc calltip      json          socket
_compat_pickle calltip_w   keyword       socketserver
_compression   cgi         lib2to3       sqlite3
_contextvars   gitb        linecache     squeezer
_csv           chunk       locale        sre_compile
_ctypes        cmath       logging       sre_constants
_ctypes_test   cmd         lzma          sre_parse
_datetime      code        macosx        ssl
_decimal       codecontext mailbox        stackviewer
_dummy_thread  codecs      mailcap       stat
_elementtree  codeop     mainmenu      statistics
_functools    collections marshal       statusbar
_hashlib       colorizer  math          string
_heapq         colorsys   mimetypes     stringprep
_imp          compileall mmap          struct
_io           concurrent modulefinder  subprocess
_json         config     msilib        sunau

```

```

help> keywords

Here is a list of the Python keywords. Enter any keyword to get more help.

False          class          from           or
None           continue      global        pass
True           def           if            raise
and            del           import       return
as            elif          in           try
assert        else         is           while
async         except        lambda        with
await         finally      nonlocal     yield
break         for          not

help> |

```

נניח ונרצה לדעת יותר על המושג "False", נכתוב מושג הזה לאחר >help ונקבל הסבר עליו.

```

help> False
Help on bool object:

class bool(int)
| bool(x) -> bool
|
| Returns True when the argument x is true, False otherwise.
| The builtins True and False are the only two instances of the class bool.
| The class bool is a subclass of the class int, and cannot be subclassed.
|
| Method resolution order:
|   bool
|   int
|   object
|
| Methods defined here:
|
| __and__(self, value, /)
|   Return self&value.
|
| __or__(self, value, /)
|   Return self|value.
|
| __rand__(self, value, /)
|   Return value&self.
|
| __repr__(self, /)
|   Return repr(self).
|
| __ror__(self, value, /)
|   Return value|self.
|
| __rxor__(self, value, /)
|   Return value^self.
|
| __xor__(self, value, /)
|   Return self^value.
|
|-----

```

כדי לצאת מהעזרה פשוט נלחץ "Enter" ונקבל הודעה הבאה:

```
help>
```

```

You are now leaving help and returning to the Python interpreter.
If you want to ask for help on a particular object directly from the
interpreter, you can type "help(object)". Executing "help('string')"
has the same effect as typing a particular string at the help> prompt.
>>>

```

## מערכת PyCharm.

### מערכת לכתיבת סקריפטים.

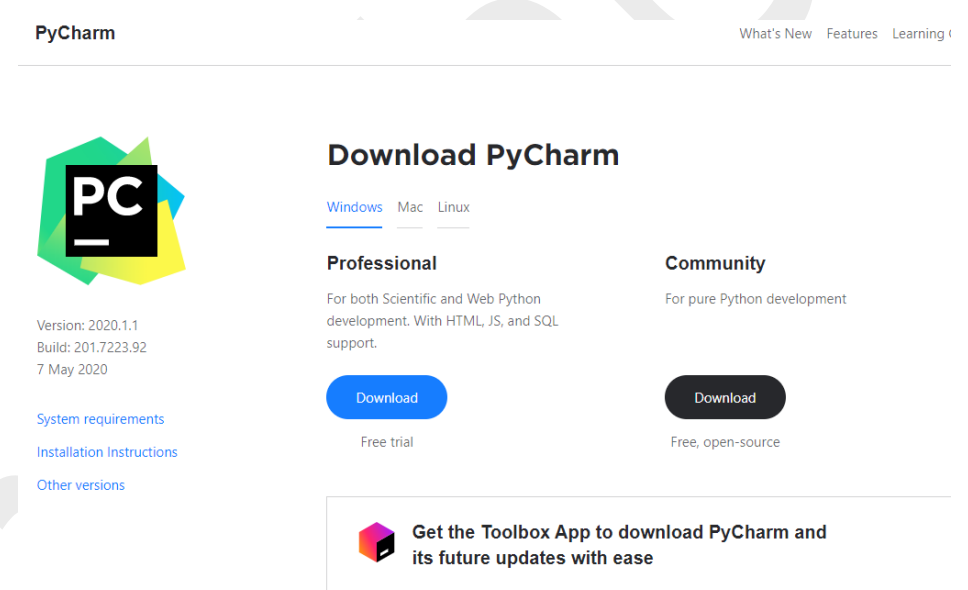
הורדת תוכנה נוחה יותר לכתיבת קוד בשפת פייתון.

ישנם מערכות שמאפשרות לכתוב קוד שלם ולקמפל אותו ואז להריץ אותו מהתחלה עד הסוף.

אחד התוכנות היא תוכנת PyCharm. ניתן להוריד אותה מהקישור הבא:

<https://www.jetbrains.com/pycharm/download/#section=windows>

יש גרסה בתשלום שמאפשרת פיתוח בסביבות שונות, ויש אפשרות לתוכנה חינמית (מה שאנחנו צריכים) שהיא תומכת רק בפייתון, ואותה נוריד. שימו לב שאתם מורידים את גרסת "Community" ולא את גרסת "Professional".



PyCharm

What's New Features Learning

### Download PyCharm

Windows Mac Linux

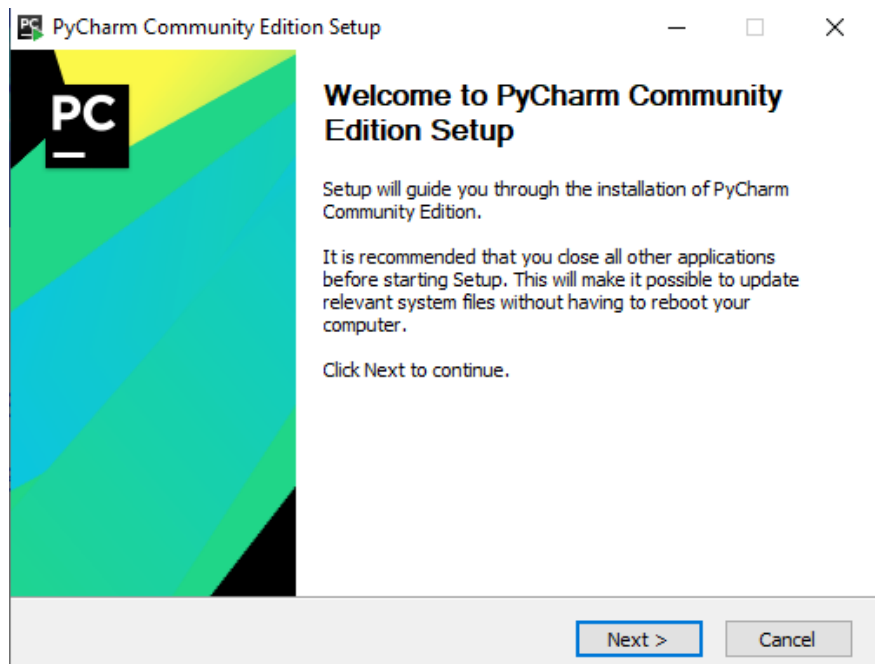
**Professional**  
For both Scientific and Web Python development. With HTML, JS, and SQL support.  
[Download](#)  
Free trial

**Community**  
For pure Python development.  
[Download](#)  
Free, open-source

Version: 2020.1.1  
Build: 201.7223.92  
7 May 2020

[System requirements](#)  
[Installation Instructions](#)  
[Other versions](#)

Get the Toolbox App to download PyCharm and its future updates with ease



PyCharm Community Edition Setup

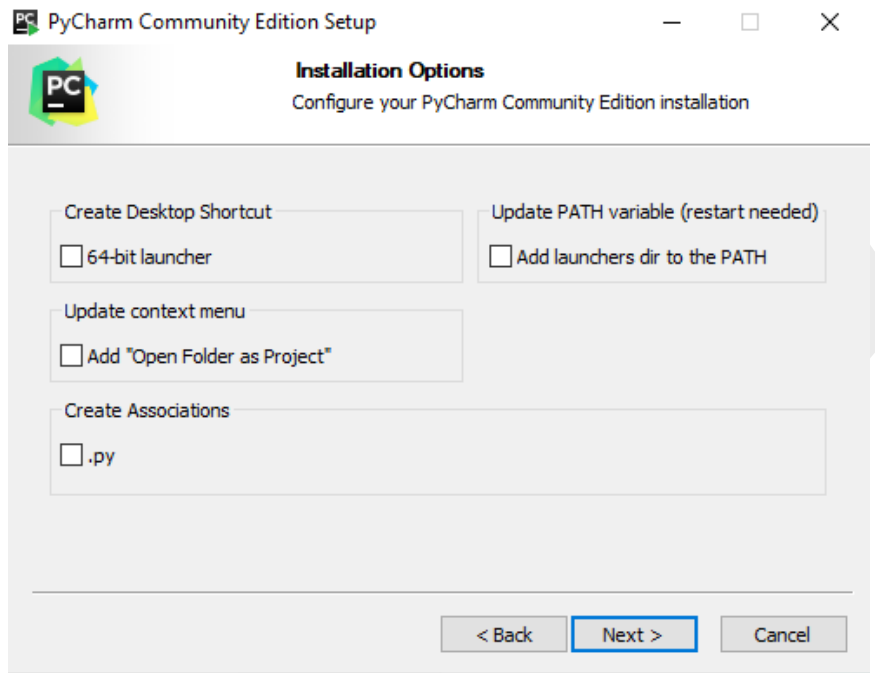
### Welcome to PyCharm Community Edition Setup

Setup will guide you through the installation of PyCharm Community Edition.

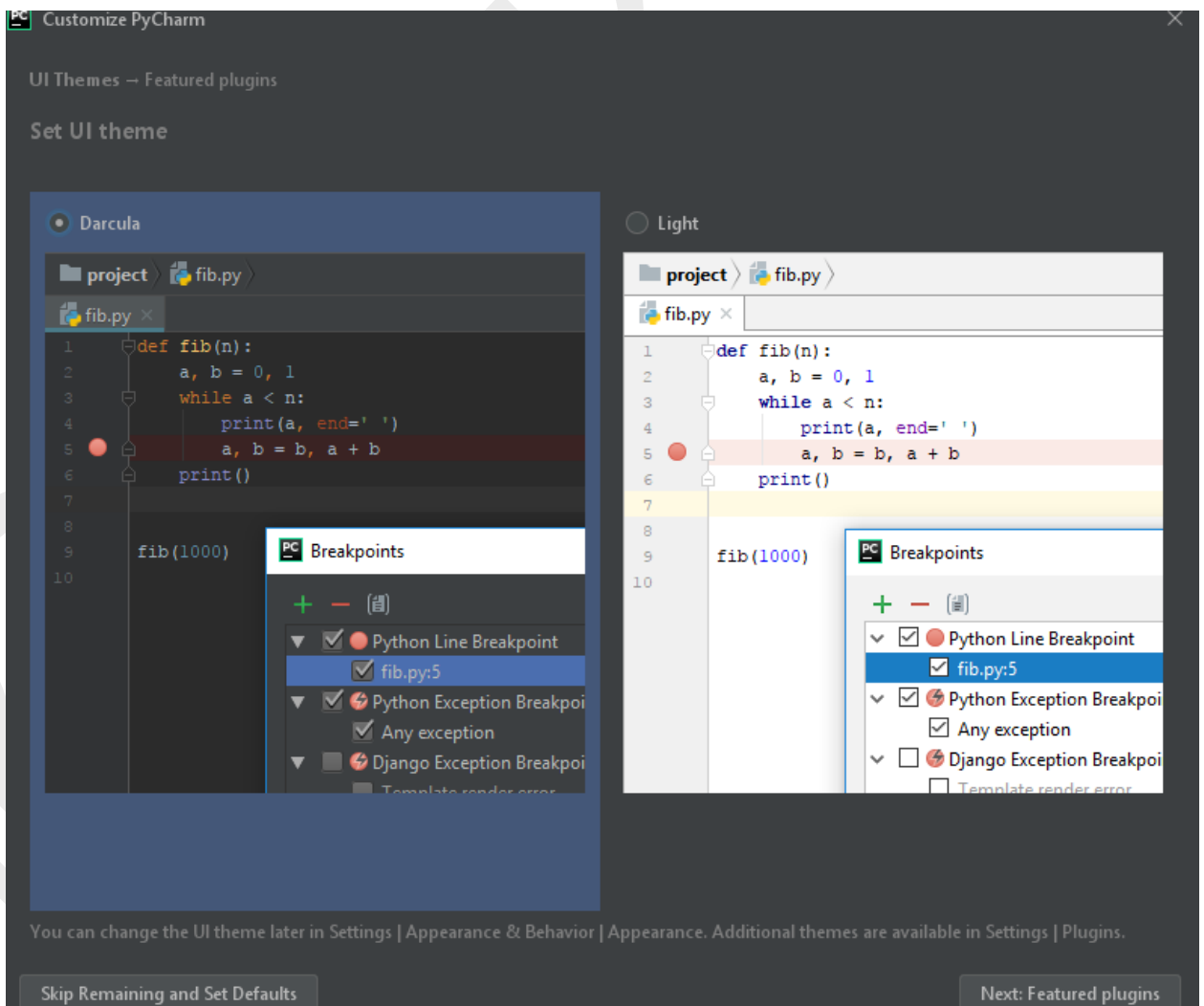
It is recommended that you close all other applications before starting Setup. This will make it possible to update relevant system files without having to reboot your computer.

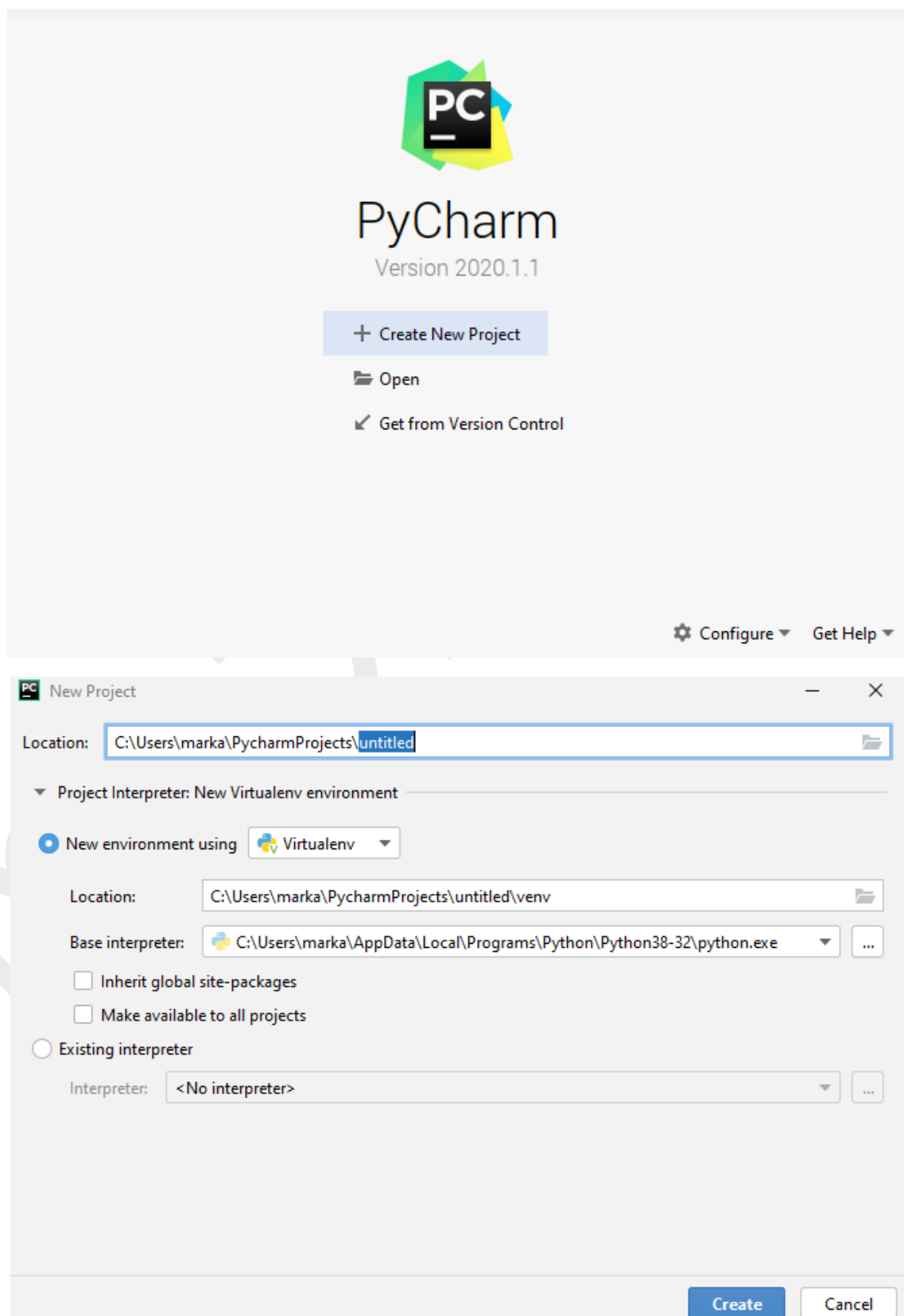
Click Next to continue.

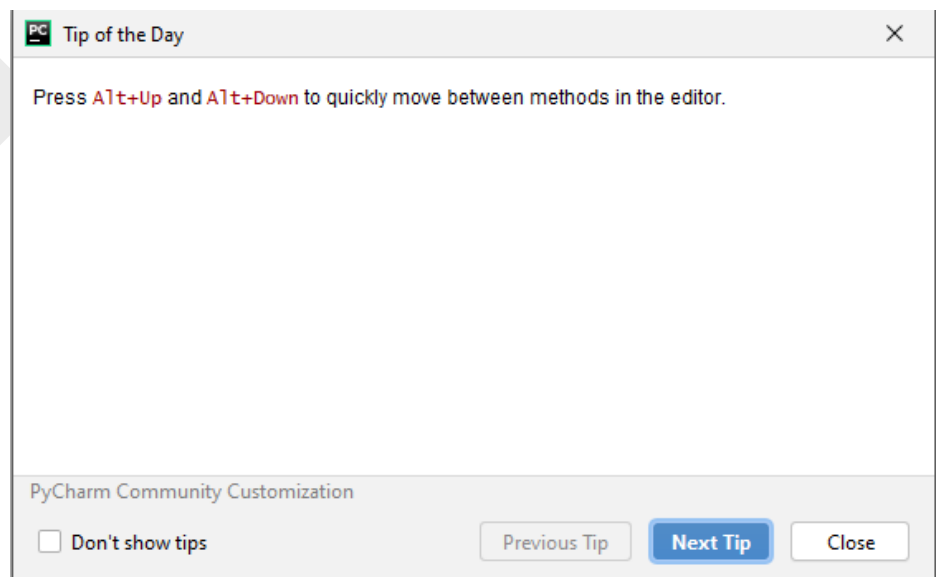
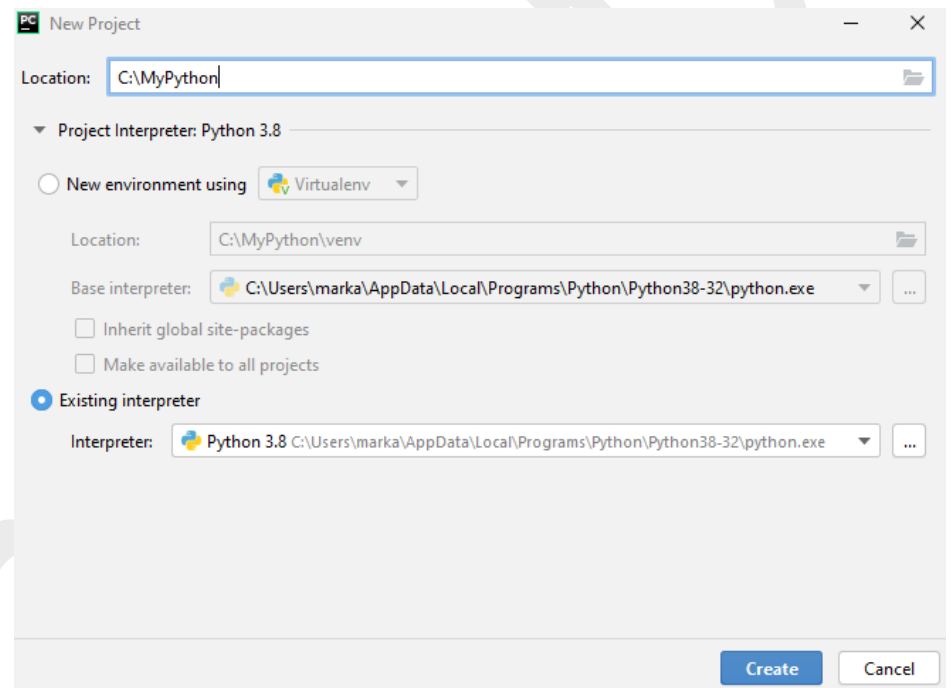
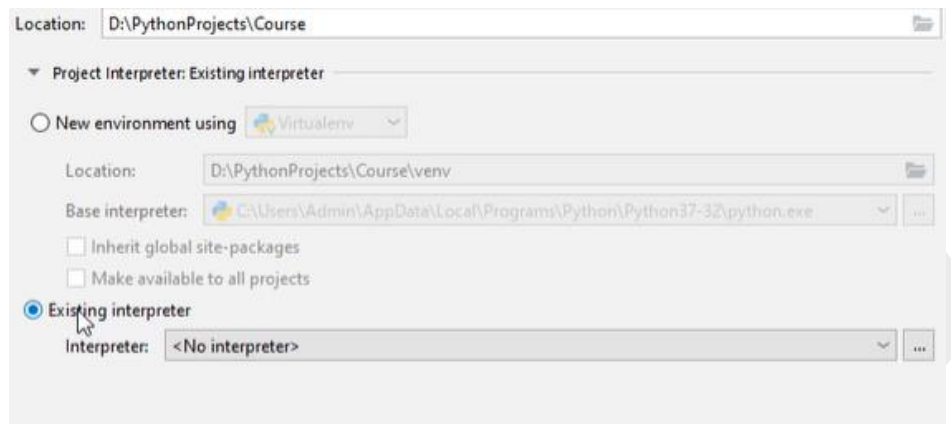
[Next >](#) [Cancel](#)

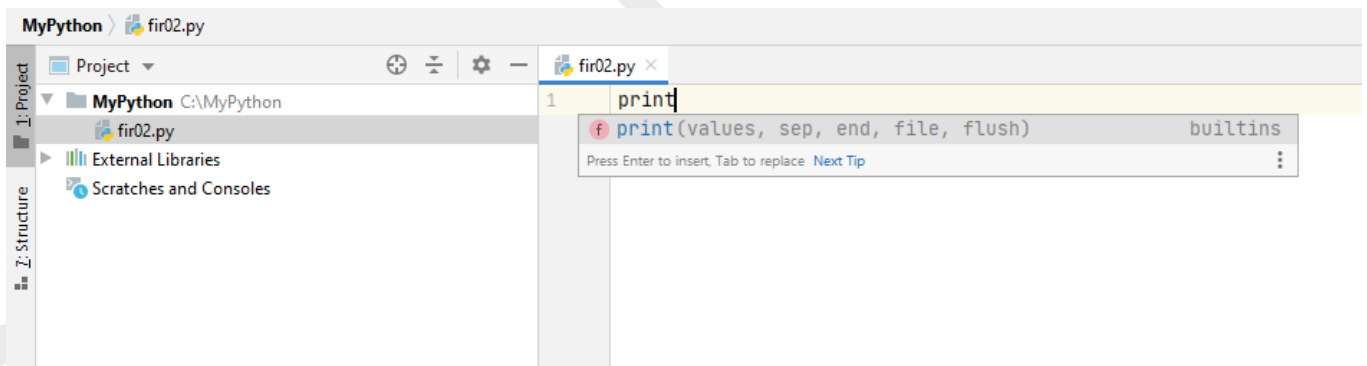
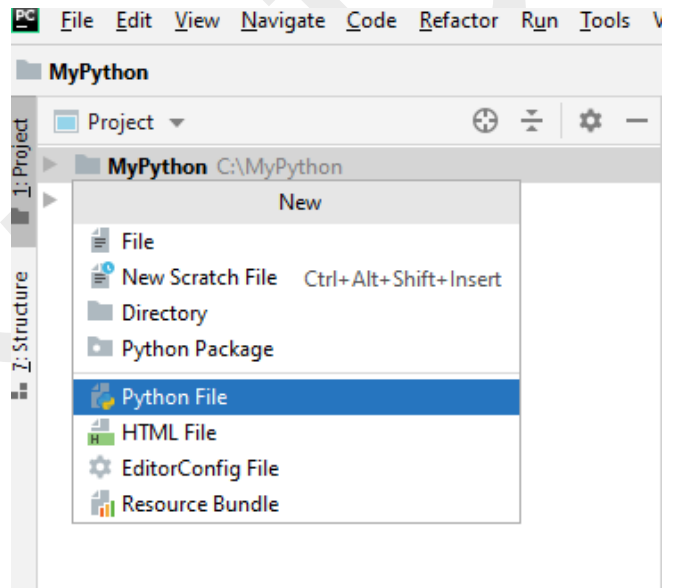
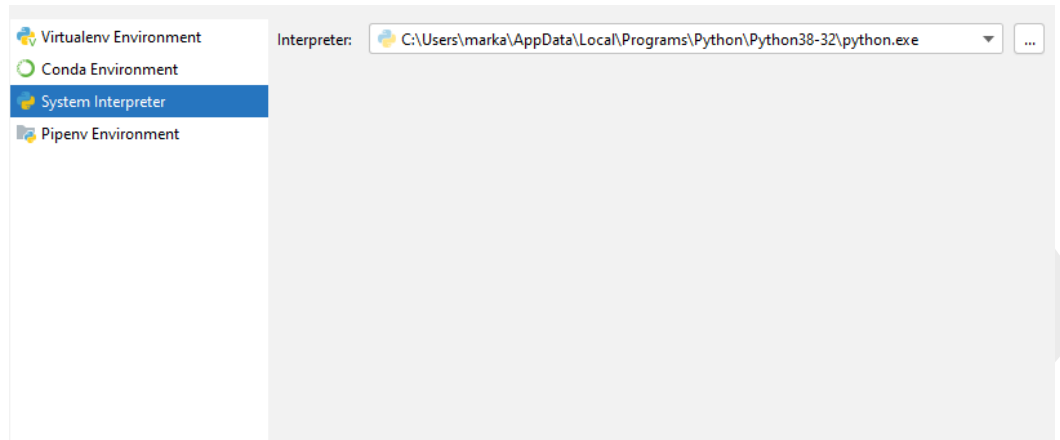


בוחרים צבע לרקע של התוכנה.

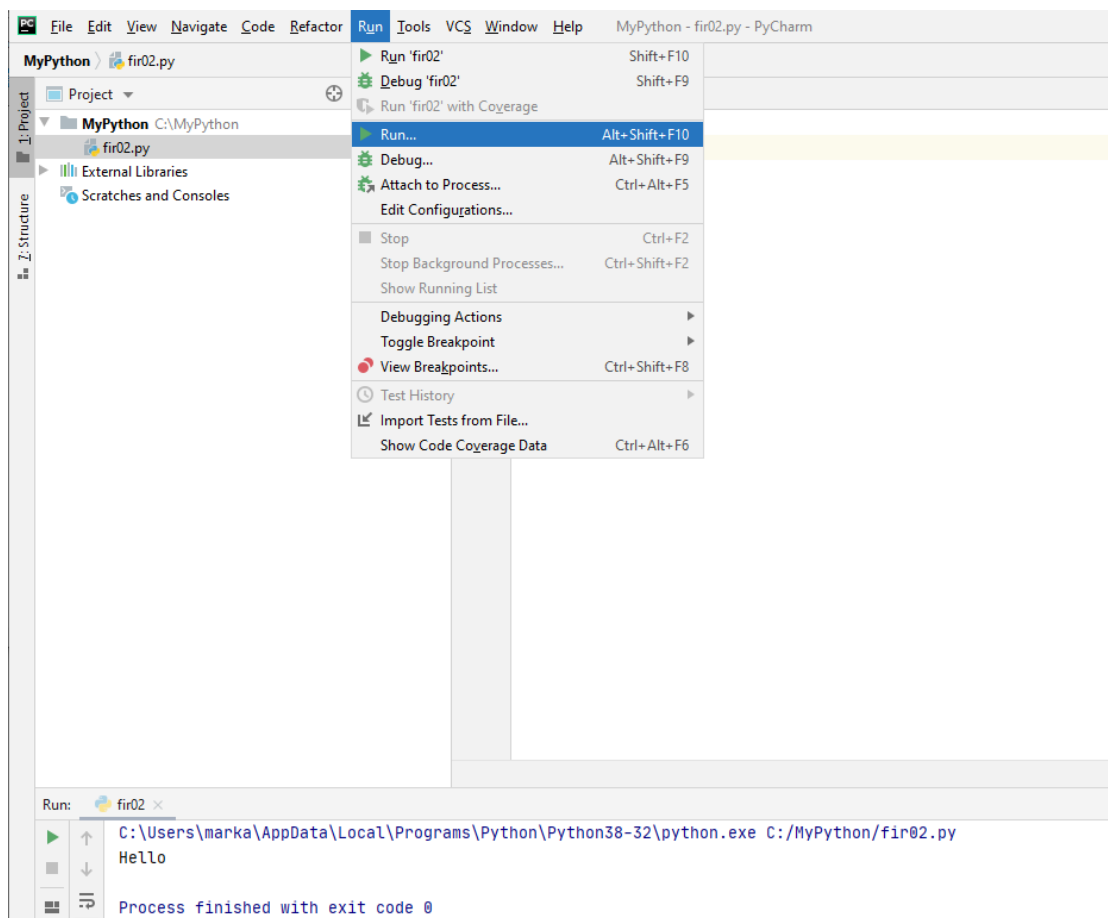




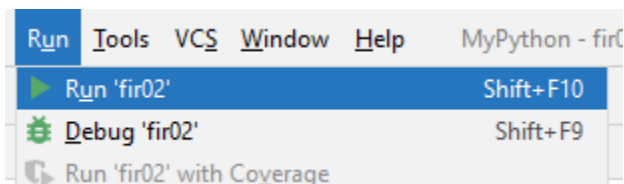








ניתן להפעיל את התוכנית בצירוף המקשים **Alt+Shift+F10**.  
לאחר שהפעלתם את הקובץ לפחות פעם אחת, קיימת אפשרות להפעיל אותו ב- **Shift+F10** או:



פעולה זו מפעילה רק את הקובץ האחרון שהפעלתם, וזה חשוב ברגע שאתם עובדים עם מספר קבצים בו זמנית.

כדי לשכפל שורה שמים סימן על שורה רצויה ולוחצים **Ctrl + d**

```
print("Hello")
print("Hello")
print("Hello")
print("Hello")
print("Hello")
```

נראה איך פייתון עובד עם שפות שונות:

```
MyPython > fir02.py
Project
  MyPython C:\MyPython
    fir02.py
  External Libraries
  Scratches and Consoles
Z: Structure

1 print("Hello")
2 print("אני כאן")
3 print("Я здесь")
4
5

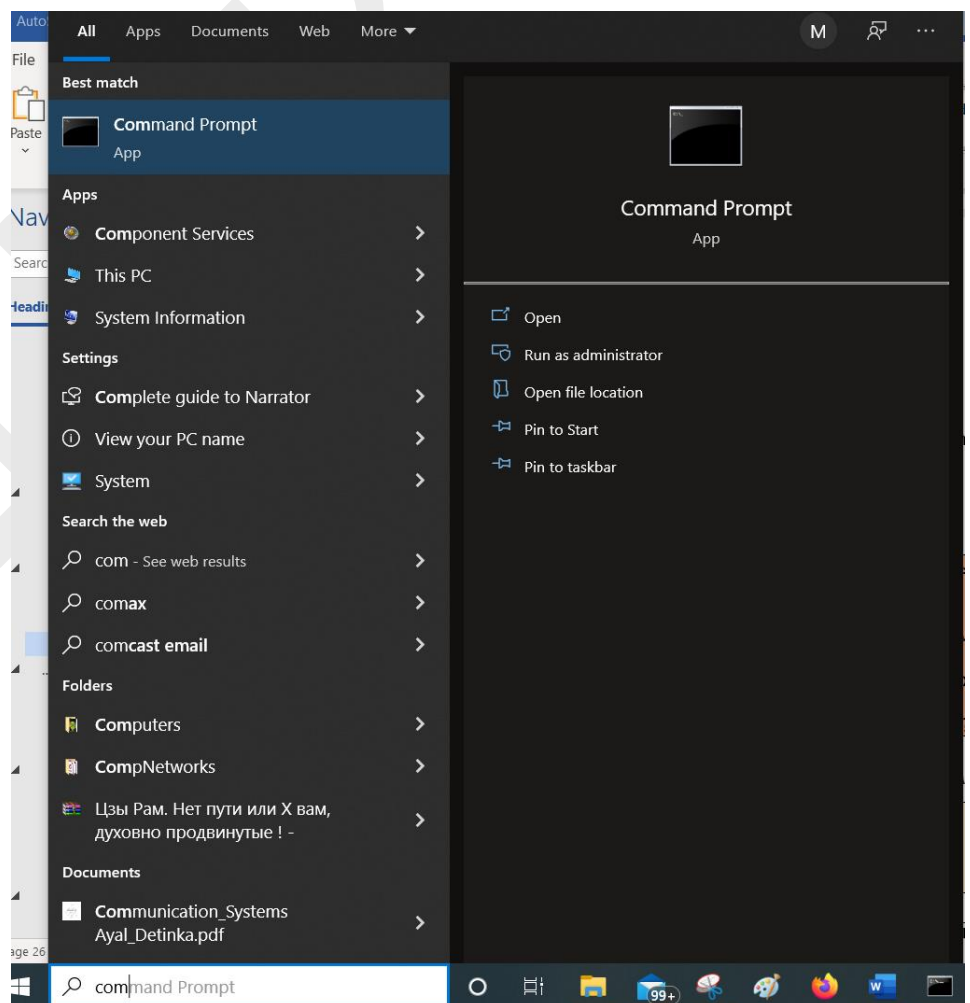
Run: fir02 x
C:\Users\marka\AppData\Local\Programs\Python\Python38-32\python.exe C:/MyPython/fir02.py
Hello
אני כאן
Я здесь
```

עוד פונקציה נוחה של התוכנה: אם רוצים להוריד או להעלות את השורה אני מצביע על השורה שאני רוצה להעביר, לוחץ Ctrl + Shift על חץ כיוון רצוי, למעלה או למטה.

הוספת מודולים בעזרת מערכת PIP.  
מה זה PIP?



pip היא מערכת ניהול חבילות הכתובה ב-Python המשמשת להתקנה ולניהול חבילות תוכנה. הוא מתחבר למאגר מקוון של חבילות ציבוריות, הנקרא Python Package Index. ניתן גם להגדיר את pip להתחבר למאגרי חבילות אחרים, בתנאי שהם תואמים לפייתון. כדי להשתמש בניהול חבילות יש להיכנס ל-command prompt:



## התקנה ועדכון של מערכת PIP.

אתם לא צריכים להתקין PIP במקרה שאתם עובדים בסביבה ווירטואלית או במקרה שהורדתם פיתון דרך האתר [./https://www.python.org](https://www.python.org).

בפעם ראשונה שאתם רוצים להתקין את המערכת יש להיכנס ולהוריד את הקובץ לפי קישור הבא: <https://bootstrap.pypa.io/get-pip.py>

אחרי זה כנסו ל- Command Prompt, תעברו לספרייה שמכילה קובץ `get-pip.py`.  
תרשמו פקודה הבאה:

```
py get-pip.py
```

(אם אתם משתמשים במחשב עם מערכת הפעלה Windows).

אם אתם משתמשים בלינוקס או Mac-Os פקודה נראית כך:

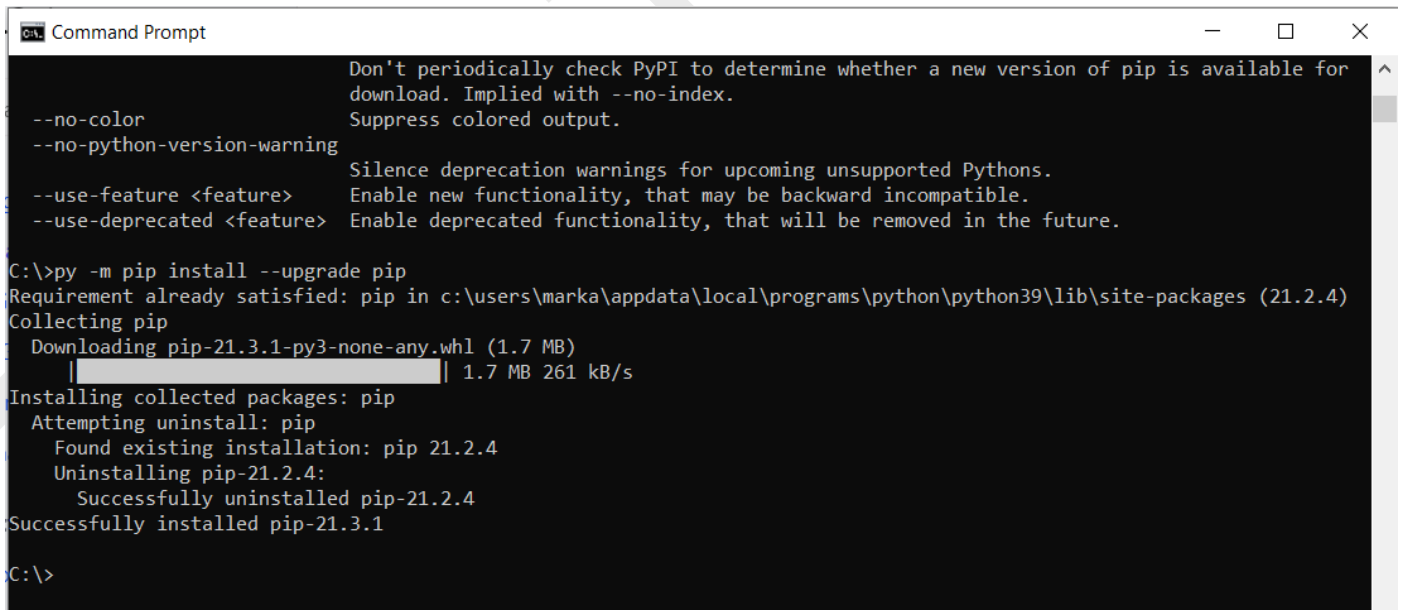
```
$ python get-pip.py
```

מידע נוסף על PIP אפשר למצוא בקישור הבא:

<https://github.com/pypa/get-pip>

מדי פעם תצטרכו לבצע חידוש גרסה לזה יש פקודה אחרת:

```
py -m pip install --upgrade pip
```



```
Command Prompt
Don't periodically check PyPI to determine whether a new version of pip is available for
download. Implied with --no-index.
--no-color Suppress colored output.
--no-python-version-warning Silence deprecation warnings for upcoming unsupported Pythons.
--use-feature <feature> Enable new functionality, that may be backward incompatible.
--use-deprecated <feature> Enable deprecated functionality, that will be removed in the future.

C:\>py -m pip install --upgrade pip
Requirement already satisfied: pip in c:\users\marka\appdata\local\programs\python\python39\lib\site-packages (21.2.4)
Collecting pip
  Downloading pip-21.3.1-py3-none-any.whl (1.7 MB)
    |-----| 1.7 MB 261 kB/s
Installing collected packages: pip
  Attempting uninstall: pip
    Found existing installation: pip 21.2.4
    Uninstalling pip-21.2.4:
      Successfully uninstalled pip-21.2.4
Successfully installed pip-21.3.1

C:\>
```

רושמים pip בתוך חלון הנפתח ומקבלים רשימת אופציות לפקודה הזאת.

```
C:\Users\marka>pip

Usage:
  pip <command> [options]

Commands:
  install           Install packages.
  download          Download packages.
  uninstall         Uninstall packages.
  freeze            Output installed packages in requirements format.
  list              List installed packages.
  show              Show information about installed packages.
  check             Verify installed packages have compatible dependencies.
  config            Manage local and global configuration.
  search            Search PyPI for packages.
  cache             Inspect and manage pip's wheel cache.
  index             Inspect information available from package indexes.
  wheel             Build wheels from your requirements.
  hash             Compute hashes of package archives.
  completion        A helper command used for command completion.
  debug            Show information useful for debugging.
  help             Show help for commands.
```

פונקציות שימושיות של PIP:  
פקודה מראה את כל החבילות (מודולים) שהותקנו במערכת:

py -m pip list

```
C:\Users\marka>py -m pip list
Package      Version
-----
future       0.18.2
iso8601      0.1.16
pip          21.3.1
py2exe       0.9.2.2
pyFirmata    1.1.0
pyserial     3.5
PyYAML       5.4.1
serial       0.0.97
setuptools   57.4.0
```

פקודה שמראה אילו חבילות מיושנות ומה הן גרסאות חדשות שנמצאות ברשת, כמובן מהאלה שמותקנות במחשב.

py -m pip list --outdated --format columns

```
C:\Users\marka>py -m pip list --outdated --format columns
Package      Version Latest  Type
-----
iso8601      0.1.16  1.0.2   wheel
py2exe       0.9.2.2 0.11.0.1 wheel
PyYAML       5.4.1   6.0     wheel
setuptools   57.4.0  60.5.0  wheel
```

אם רוצים לראות חבילות בלתי תלויות שמערכת לא חייבת (מהאלה שלא מעודכנות):

```
py -m pip list --outdated --not-required
```

```
C:\Users\marka>py -m pip list --outdated --not-required
Package      Version Latest Type
-----
py2exe       0.9.2.2 0.11.0.1 wheel
setuptools   57.4.0  60.5.0   wheel
```

אם ברצוננו לראות חבילות בלתי תלויות שמערכת לא חייבת (מסה"כ חבילות):

```
py -m pip list --outdated --not-required
```

```
C:\Users\marka>py -m pip list --not-required
Package      Version
-----
pip          21.3.1
py2exe       0.9.2.2
pyFirmata    1.1.0
serial       0.0.97
setuptools   57.4.0
```

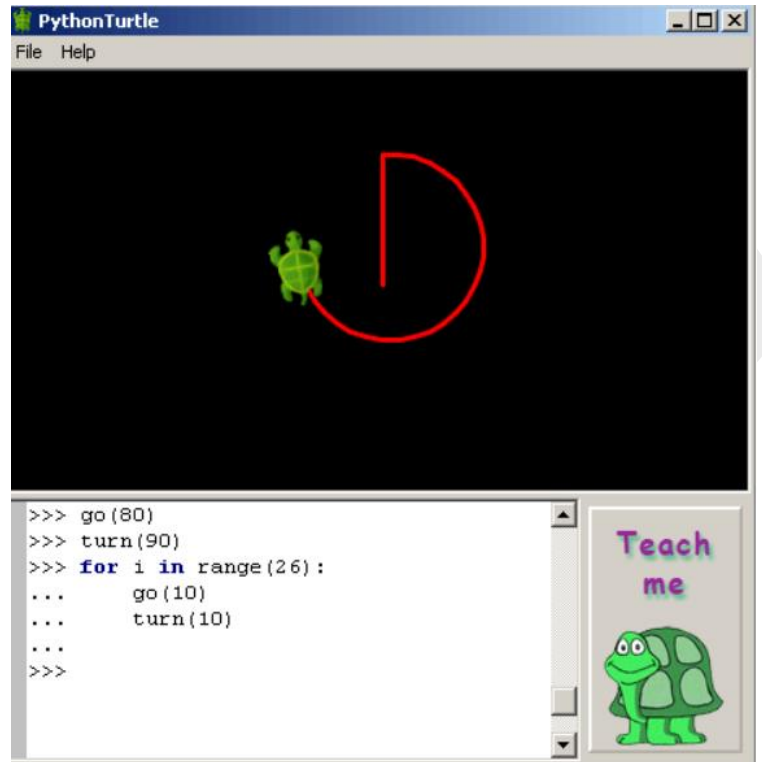
אם ברצוננו לראות את כל החבילות לא בצורת טבלה משתמשים בפקודה:

```
C:\> py -m pip list --format=json
```

```
C:\Users\marka>py -m pip list --format=json
[{"name": "future", "version": "0.18.2"}, {"name": "iso8601", "version": "0.1.16"}, {"name": "pip", "version": "21.3.1"}, {"name": "py2exe", "version": "0.9.2.2"}, {"name": "pyFirmata", "version": "1.1.0"}, {"name": "pyserial", "version": "3.5"}, {"name": "PyYAML", "version": "5.4.1"}, {"name": "serial", "version": "0.0.97"}, {"name": "setuptools", "version": "57.4.0"}]
```

התקנת מודולים (חבילות):

נניח כי ברצוננו להתקין חבילה של "צב" (Turtle).



אם ניכנס לCMD ונרשום "PythonTurtle", נקבל:

```
C:\Users\marka>PythonTurtle
'PythonTurtle' is not recognized as an internal or external command,
operable program or batch file.
```

אנו צריכים להתקין את החבילה:

```
C:\Users\marka>pip install PythonTurtle
Collecting PythonTurtle
  Downloading PythonTurtle-0.3.2-py3-none-any.whl (518 kB)
    |-----| 518 kB 105 kB/s
Installing collected packages: PythonTurtle
Successfully installed PythonTurtle-0.3.2
```

גם אחרי זה אם נריץ תוכנה נקבל שגיאה, כי חסר עוד מודולים:

```
C:\Users\marka>PythonTurtle
wxPython doesn't seem to be installed. You need to install the appropriate prerequisites for your operating system.
Please consult the installation instructions in the README at https://github.com/PythonTurtle/PythonTurtle#installation
```

אם נתקין מודול wxPython אז PIP תתקין גם חבילה הזאת וגם מספר חבילות נוספות.

```
C:\Users\marka>pip install wxPython
Collecting wxPython
  Downloading wxPython-4.1.1-cp39-cp39-win_amd64.whl (18.1 MB)
    |████████████████████████████████████████| 18.1 MB 93 kB/s
Collecting numpy
  Downloading numpy-1.22.0-cp39-cp39-win_amd64.whl (14.7 MB)
    |████████████████████████████████████████| 14.7 MB 409 kB/s
Collecting six
  Downloading six-1.16.0-py2.py3-none-any.whl (11 kB)
Collecting pillow
  Downloading Pillow-9.0.0-cp39-cp39-win_amd64.whl (3.2 MB)
    |████████████████████████████████████████| 3.2 MB 467 kB/s
Installing collected packages: six, pillow, numpy, wxPython
Successfully installed numpy-1.22.0 pillow-9.0.0 six-1.16.0 wxPython-4.1.1
```

יכול להיות מצב שנקבל שגיאת התקנה:

The screenshot shows a Windows desktop environment. In the background, a web browser window displays the Microsoft Visual Studio page for downloading build tools. In the foreground, a Command Prompt window shows the following error output:

```
VCInstallDir
raise distutils.errors.DistutilsPlatformError(msg)
distutils.errors.DistutilsPlatformError: Microsoft Visual C++ 14.2 or greater is required. Get it with "Microsoft
C++ Build Tools": https://visualstudio.microsoft.com/visual-cpp-build-tools/
Finished command: build_wx (0m2.167s)
Finished command: build (0m2.167s)
WARNING: Building this way assumes that all generated files have been
generated already. If that is not the case then use build.py directly
to generate the source and perform the build stage. You can use
--skip-build with the bdist_* or install commands to avoid this
message and the wxWidgets and Phoenix build steps in the future.

"C:\Users\natik\AppData\Local\Programs\Python\Python310\python.exe" -u build.py build
Command "'C:\Users\natik\AppData\Local\Programs\Python\Python310\python.exe" -u build.py build' failed with exit c
ode 1.
[end of output]
note: This error originates from a subprocess, and is likely not a problem with pip.
error: legacy-install-failure
x Encountered error while trying to install package.
  wxpython
note: This is an issue with the package mentioned above, not pip.
hint: See above for output from the failure.

C:\Users\natik\Downloads>
C:\Users\natik\Downloads>https://visualstudio.microsoft.com/visual-cpp-build-tools/
```

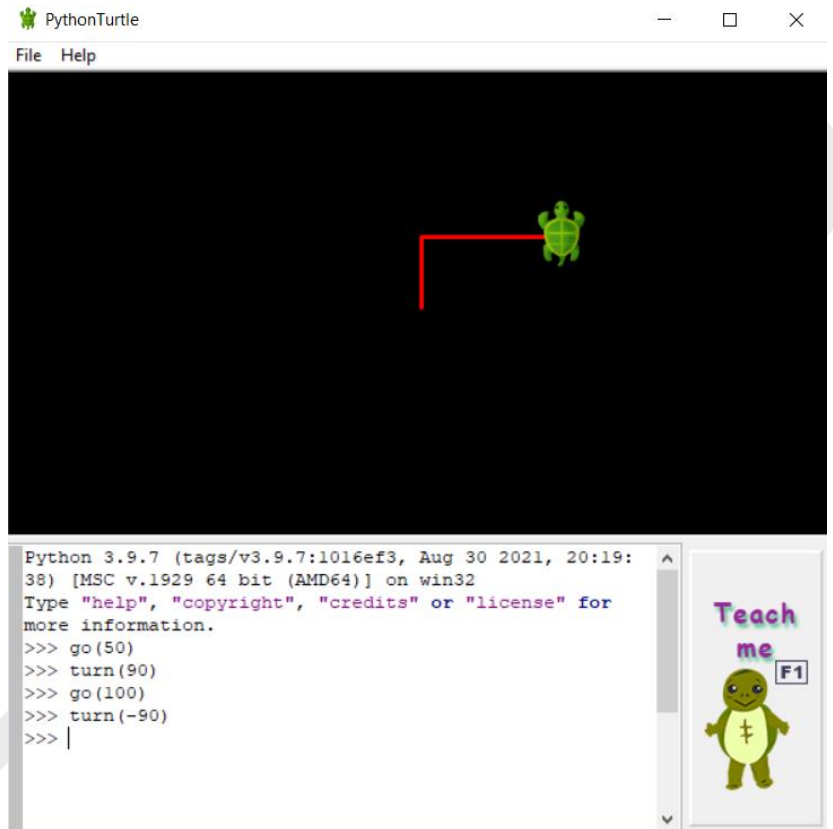
בשביל לפתור בעיה זו יש להתקין Visual Studio Tools לפי קישור הבא:

<https://visualstudio.microsoft.com/visual-cpp-build-tools/>



C:\Users\marka>PythonTurtle :אם אחרי זה נריץ את הפקודה:

נקבל:



לאחר התקנה, אם נבדוק את החבילות נקבל:

```
C:\Users\marka>py -m pip list
Package          Version
-----
future           0.18.2
iso8601          0.1.16
numpy            1.22.0
Pillow           9.0.0
pip              21.3.1
py2exe           0.9.2.2
pyFirmata        1.1.0
pyserial         3.5
PythonTurtle    0.3.2
PyYAML           5.4.1
serial           0.0.97
setuptools       57.4.0
six              1.16.0
wxPython         4.1.1
```

כל הפקודות שקשורות לנושא PIP נמצאות לפי כתובת:  
[/https://pip.pypa.io/en/stable/cli/pip\\_download](https://pip.pypa.io/en/stable/cli/pip_download)

נראה עוד דוגמה של Turtle:

```
import time
import turtle
from turtle import *
#turtle.up()
turtle.ht()
turtle.goto(80,80)
turtle.st()
turtle.bgcolor("black")
turtle.color("cyan")
turtle.shape("turtle")
turtle.speed(3)
turtle.pensize(10)
turtle.right(-90)
turtle.fd(130)
turtle.left(140)
turtle.fd(30)
penup()
turtle.ht()
turtle.goto(200,85)
pendown()
turtle.st()
def draw(rad):
    # rad --> radius of arc
    for i in range(2):
        # two arcs
        turtle.circle(rad, 90)
        turtle.circle(rad // 2, 90)
# Main section
# tilt the shape to negative 45
turtle.seth(45)
# calling draw method
draw(75)
time.sleep(10)
```

נראה את התמונה הבאה:



פונקציה print – הדפסת נתונים על המסך (או בשלב מתקדם לתוך הקובץ).

שימו לב שבשפת פייתון אפשר לכתוב טקסט להדפסה גם בין גרשיים וגם בין "גרש - גרש".

```
print("אני כאן")  
print("Я здесь")  
print("Hello")  
print('Hello')
```

```
C:\Users\marka\AppData\Local\Programs\Python\Python38-32\python.exe C:/MyPython/third.py  
אני כאן  
Я здесь  
Hello  
Hello
```

בברירת מחדל לאחר הדפסה של מחרוזת או משתנה נקבל ירידת שורה - \n נכנס באופן אוטומטי.  
הדפסה של מספר שדות:

```
a=5  
b=7  
print("a=", a, "b=", b)  
print("a+b=",a+b)
```

```
a= 5 b= 7
a+b= 12
```

בסוגרים אפשר להכניס סוגים שונים של נתונים וכמות כלשהי של הנתונים:

```
>>> print(12, 12.3, "Go West", 0x123)
12 12.3 Go West 291
```

ניתן להדפיס גם משתנים וגם ערכים קבועים:

```
>>> a=13
>>> b=17.5
>>> c= "We are the champions"
>>> print( "1- ", a, "2 -", b, "3- ", c)
1- 13 2 - 17.5 3- We are the champions
```

**פונקציה print() למתקדמים:**

פונקציה print היא פונקציה הרבה יותר מורכבת מסתם הדפסה:

```
print(*objects, sep=' ', end='n', file=sys.stdout, flush=False)
```

\*objects – זה אומר שיכול להיות מספר אובייקטים בהדפסה.

' sep=' – זה אומר שהפרדה בין אובייקטים בברירת מחדל היא סימן רווח, אבל אתם יכולים לקבוע גם סימן אחר.

'end='n' – שמים לאחר כל האובייקטים.

file=sys.stdout במידה ורוצים לשלוח קובץ למסך או לקובץ.

flush=False בברירת מחדל FALSE מעביר תוכן הודעה למסך, אם ערך TRUE מעבירים אותו לקובץ.

כל המילים הנ"ל ( sep , flush , file , end ) הן מילים שמורות ואם רוצים להשתמש בשדה מיוחד חייבים להשתמש במילים האלה.

לדוגמה:

```
print("Go", "Went", "Gone", sep = "-")
```

פלט:

```
Go-Went-Gone
```

שימוש בפרמטר END. שימו לב לדוגמה:

```
a=5
b=7
print("a=", a, "b=", b)
print("a+b=",a+b,end='\n')
```

```
print("a+b=",a+b,end=' ')
print("a+b=",a+b,end='+')
print("a+b=",a+b,end='\t')
print("a+b=",a+b,end='\v')
```

שימו לב שסימנים מיוחדים שאנו רגילים בשפת C – '\t' ' ' \v לא עובדים כאן.  
ולפלט שלה:

```
C:\Users\marka\AppData\Local\Programs\Python\Python38-32\python.exe C:/MyPython/fir02.py
a= 5 b= 7
a+b= 12
a+b= 12 a+b= 12+a+b= 12 a+b= 12
Process finished with exit code 0
```

אם לא משנים פרמטר זה, אז הפונקציה באופן אוטומטי עוברת לשורה חדשה. סימנות שראינו בשפת C לא עובדות כאן, חוץ מ- "\n".

בשפת פייתון יש 2 אפשרויות להדפסה:

סידור בפורמט הישן (C style).

```
>>> pupil = "Ben"
>>> old=16
>>> grade=9.2
>>> print("It is the c Style %s, %d, %f" % (pupil, old, grade))
It is the c Style Ben, 16, 9.200000
```

כמו בשפת C %s כדי להדפיס מחרוזת %c כדאי להדפיס תו- %d – מספר שלם- %f – מספר ממשי.  
במידה ורוצים לקצץ את כמות האפסים ניתן להשתמש בצורה דומה לשפת C:

```
>>> print("It's %s, %d. Level: %.1f" % (pupil, old, grade))
It's Ben, 16. Level: 9.2
```

סידור בעזרת format.

```
>>> print("This is a {0}. It's {1}.".format("ball", "red"))
This is a ball. It's red.
>>> print("This is a {0}. It's {1}.".format("cat", "white"))
This is a cat. It's white.
>>> print("This is a {0}. It's {1} {2}.".format(1, "a", "number"))
This is a 1. It's a number.
```

בסוגריים מסולסלות נמצאים מספרים של הנתונים שנמצאים לאחר המילה השמורה: "format".

```
>>> print("This is a {0}. It's {1}.".format("ball", "red"))
This is a ball. It's red
>>> print("This is a {0}. It's {1}.".format("cat", "white"))
This is a cat. It's white
>>> print("This is a {0}. It's {1} {2}.".format(1, "a", "number"))
This is a 1. It's a number.
```

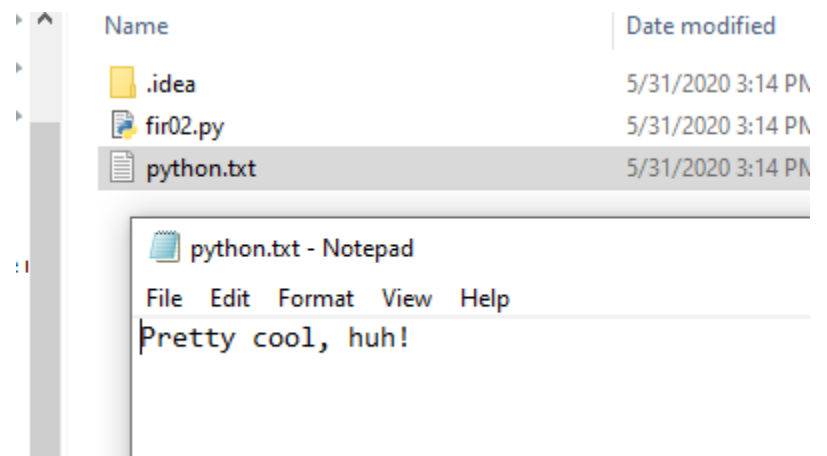
להבדיל משפת C הפונקציה הזאת לא מחזירה ערך.

```
>>> a=print("123")
123
>>> print(a)
None
```

שמירת נתונים לתוך הקובץ:

```
sourceFile = open('python.txt', 'w')
print('Pretty cool, huh!', file = sourceFile)
sourceFile.close()
```

שימו לב שצורת הכתיבה דומה לצורה בשפת C – פותחים את הקובץ עם attribute מיוחד, מעתיקים לתוכו את תוכן הקובץ ואז סוגרים (משחררים) את הקובץ.



## משתנים, מספרים ופעולות.

### אין צורך להגדיר סוג המשתנה בשפת פייתון.

שם המשתנה יכול להכיל אותיות באנגלית (גם קטנות וגם גדולות), ספרות מ-0 עד 9 וגם סימן "\_". סימן ראשון בשם המשתנה חייב להיות אות. מומלץ לתת שמות לוגיים שמשקפים את השימוש והמטרה של המשתנה. יש סדרת שמות (מילים שמורות reserved words) שאי-אפשר להשתמש בהם לצורך הגדרת שם המשתנה:

and	except	lambda	with
as	finally	nonlocal	while
assert	false	None	yield
break	for	not	
class	from	or	
continue	global	pass	
def	if	raise	
del	import	return	
elif	in	True	
else	is	try	

שימו לב שיש הבדל בין אותיות גדולות וקטנות כמו בשפת C: משתנה kuku – זה לא אותו משתנה כמו KUKU:

```
>>> d=100
>>> print(d)
100
>>> print(D)
Traceback (most recent call last):
  File "<pyshell#16>", line 1, in <module>
    print(D)
NameError: name 'D' is not defined
```

צורת ההשמה המיוחדת שיש בפייתון:

```
>>> c=a=b=1
>>> print(c,a,b)
1 1 1
```

באנגלית משתנה נקרא variable וערך שלו נקרא literal.

בפייתון ישנם 3 סוגים של משתנים נומריים: שלם (int), ממשי (float), מדומה (complex).

```
A=10
print(a)
a=1
print(a)
```

```
num=5.5
print(num)
num=1.52e2
print(num)
a=3j
print(a)
a=2-3j
print(a)
num=1.5j
print(num)
```

```
C:\Users\marka\AppData\Local\Programs\Python\Python38-32\python.exe C:/MyPython/fir02.py
10
1
5.5
152.0
3j
(2-3j)
1.5j
Process finished with exit code 0
```

ניתן להגדיר מספרים בבסיס שונה מבסיס 10.

```
A=0b10010110
print(a)
a=0o310
print(a)
num=0x21c
print(num)
num=0X123
print(num)
num=0X12C
print(num)
num=0x12C
print(num)
```

```
C:\Users\marka\AppData\Local\Programs\Python\Python38-32\python.exe C:/MyPython/fir02.py
150
200
540
291
300
300
```



שימו לב שפייתון זאת שפה מופשטת, שמאפשרת לרשום מספרים בבסיס 16 גם באותיות קטנות וגם באותיות גדולות.

### פעולת החלפת נתונים:

בפייתון קיימת אפשרות להחלפת ערכים בין שני משתנים.

בשפות אחרות יש צורך להשתמש במשתנה שלישי "temp" כדי לעשות את זה:

```
temp=b;
b=a;
a=temp;
```

בפייתון ניתן לבצע פעולה זו בצורה יותר קלה:

```
>>> a='q'
>>> b=3
>>> (a,b) = (b,a)
>>> print(a,b)
3 q
>>> a,b=b,a
>>> print(a,b)
q 3
```

שימו לב שניתן להחליף גם משתנים מסוגים שונים. במקרה כזה גם משתנים הסוגים של המשתנים:

```
>>> c='Name'
>>> d=3.2
>>> typeI
<class 'str'>
>>> type(d)
<class 'float'>
>>> print(c,d)
Name 3.2
>>> c,d=d,c
>>> printI
3.2
>>> print(d)
Name
>>> typeI
<class 'float'>
>>> type(d)
<class 'str'>
```

## משתנים מסוג מחרוזת – String.

ניתן להגדיר מחרוזת גם בין גרש לגרש ('), וגם בין זוג גרשיים לזוג גרשיים (") ואפילו בין שלישיית גרשיים ("""). לצורה "" יש משמעות מיוחדת – במידה ומגדירים מחרוזת במספר שורות – משתמשים בצורה הזאת.

```
A='0b10010110'  
print(a)  
a="0o310"  
print(a)  
num=""0x21c  
print(num)  
num=0X123  
print(num)  
num=0X12C"  
print(num)
```

```
C:\Users\marka\AppData\Local\Programs\Python\Python38-32\python.exe C:/MyPython/fir02.py  
0b10010110  
0o310  
0x21c  
print(num)  
num=0X123  
print(num)  
num=0X12C
```

### פעולות עם מחרוזות:

ניתן לחבר מחרוזות למחרוזת אחת, ניתן להכפיל את המחרוזת במספר קבוע ואז נשכפל אותה מספר פעמים. נראה את הדוגמה:

```
>>> a='QWERTY'  
>>> b="CBW"  
>>> d=a+b  
>>> print(d)  
QWERTYCBW  
>>> print(a + ' ' + b)  
QWERTY CBW  
>>> d=a*3  
>>> print(d)  
QWERTYQWERTYQWERTY
```

אין אפשרות להשתמש בפעולות חיסור וחילוק בין מחרוזות. אי-אפשר להוסיף למחרוזת מספר, אי-אפשר להכפיל את המחרוזת בתו.

```
>>> d=a-b
Traceback (most recent call last):
  File "<pyshell#7>", line 1, in <module>
    d=a-b
TypeError: unsupported operand type(s) for -: 'str' and 'str'
>>> d=a/3
Traceback (most recent call last):
  File "<pyshell#8>", line 1, in <module>
    d=a/3
TypeError: unsupported operand type(s) for /: 'str' and 'int'
>>> d=a-3
Traceback (most recent call last):
  File "<pyshell#9>", line 1, in <module>
    d=a-3
TypeError: unsupported operand type(s) for -: 'str' and 'int'
>>> d=a-'3'
Traceback (most recent call last):
  File "<pyshell#10>", line 1, in <module>
    d=a-'3'
TypeError: unsupported operand type(s) for -: 'str' and 'str'
>>> d=a+3
Traceback (most recent call last):
  File "<pyshell#11>", line 1, in <module>
    d=a+3
TypeError: can only concatenate str (not "int") to str
>>> d=a*'3'
Traceback (most recent call last):
  File "<pyshell#12>", line 1, in <module>
    d=a*'3'
TypeError: can't multiply sequence by non-int of type 'str'
```

פונקציות נוספות עם תווים ומחרוזות:  
:Unicode מקבלת תו ומחזירה ערך של תו לפי טבלת Unicode:

```
>>> ord('z')
122
>>> ord('')
96
>>> ord('כ')
1499
>>> ord(2)
Traceback (most recent call last):
  File "<pyshell#23>", line 1, in <module>
    ord(2)
TypeError: ord() expected string of length 1, but int found
>>> ord('sdwe')
Traceback (most recent call last):
  File "<pyshell#25>", line 1, in <module>
    ord('sdwe')
TypeError: ord() expected a character, but string of length 4 found
```

הפונקציה chr() מקבלת מספר ומחזירה תו לפי טבלת Unicode:

```
>>> chr(89)
'Y'
>>> chr(20121)
'𐤒'
>>> chr(q)
Traceback (most recent call last):
  File "<pyshell#28>", line 1, in <module>
    chr(q)
NameError: name 'q' is not defined
```

פונקציה len() מקבלת מחרוזת ומחזירה עורך של המחרוזת או אובייקט מורכב אחר (לא מספר):

```
>>> len('abc')
3
>>> s1='-----'
>>> s2='++++'
>>> len(s1)>len(s2)
True
>>> len(123)
Traceback (most recent call last):
  File "<pyshell#33>", line 1, in <module>
    len(123)
```

```
TypeError: object of type 'int' has no len()
```

```
>>> len('a')
```

```
1
```

## משתנים בוליאניים.

בשפת פייתון קיים סוג משתנים שלא היה בשפת C – משתנה בוליאני – Boolean. משתנה מקבל ערך True ו-False.

במידה ומגדירים ערך בצורה בוליאנית נקבל הודעת שגיאה, ערך שמוגדר בין הגרשיים מוגדר בתור מחרוזת.

```
A=true  
print(a)  
a=True  
print(a)  
num='True'  
print(num)  
num='true'  
print(num)
```

Traceback (most recent call last):

File "C:/MyPython/fir02.py", line 2, in <module>

a=true

NameError: name 'true' is not defined

```
a=False  
print(a)  
a=True  
print(a)  
num='True'  
print(num)  
num='true'  
print(num)
```

```
C:\Users\marka\AppData\Local\Programs\Python\Python38-32\python.exe C:/MyPython/fir02.py
```

```
False
```

```
True
```

```
True
```

```
true
```

```
Process finished with exit code 0
```

כדי לבדוק סוג של משתנה בוליאני משתמשים בפקודה type()

```
>>> a=True
>>> type(a)
<class 'bool'>
```

כמו בכל שפת תכנות True=1, False=0

```
>>> int(True)
1
>>> int(False)
0
>>> bool(1)
True
>>> bool(0)
False
>>> bool(5)
True
>>> bool(-5)
True
>>> bool(' ')
True
>>> bool("")
False
```

פעולות לוגיות עם משתנים בוליאניים.

סימנים שמקובלים בפייתון הם אותם סימנים כמו בשפות אחרות:

קטן	גדול	שווה	שונה	קטן או שווה	גדול או שווה
<	>	==	!=	<=	>=

```
>>> a=10
>>> b=5
>>> a+b>14
True
>>> a<=b+5
True
>>> a!=b
True
>>> a==b
False
```

```
>>> a=10
>>> b=5
>>> c=a==b
>>> a,b,c
(10, 5, False)
>>> print(a,b,c)
10 5 False
```

בשפת פייתון יש אופציות מיוחדות שלא קיימות בשפות אחרות:

```
>>> a==b==c
False
>>> 1<=a<=10
True
```

אופרטורים מיוחדים "any", "all":

אופרטור any מחזיר True, אם לפחות אחד מהתנאים הוא True.

```
# Since all are false, false is returned
print(any([False, False, False, False]))

# Here the method will short-circuit at the
# second item (True) and will return True.
Print(any([False, True, False, False]))

# Here the method will short-circuit at the
# first (True) and will return True.
Print(any([True, False, False, False]))
```

```
False
True
True
```

אופרטור all מחזיר True, אם כל התנאים הם True.

```
# Here all the iterables are True so all
# will return True and the same will be printed
print(all([True, True, True, True]))

# Here the method will short-circuit at the
# first item (False) and will return False.
```

```
Print(all([False, True, True, False]))
```

```
# This statement will return False, as no  
# True is found in the iterables  
print(all([False, False, False]))
```

```
True  
False  
False
```

נראה דוגמה אחרת שתאפשר לנו להבין בדיוק איך זה עובד בשלב מאוחר יותר, כאשר נלמד מה זה רשימות ולולאות.

נבדוק האם לפחות מספר אחד מהרשימה מתחלק ב-5 ללא שארית.

```
# This code explains how can we  
# use 'any' function on list  
list1 = []  
list2 = []  
  
# Index ranges from 1 to 10 to multiply  
for I in range(1, 11):  
    list1.append(4 * i)  
  
# Index to access the list2 is from 0 to 9  
for I in range(0, 10):  
    list2.append(list1[i] % 5 == 0)  
  
print('See whether at least one number is divisible by 5 in list 1=>')  
print(any(list2))
```

```
See whether at least one number is divisible by 5 in list 1=>  
True
```

נבדוק האם כל המספרים ברשימה הם מספרים אי-זוגיים.

```
# Illustration of 'all' function in python 3  
  
# Take two lists  
list1=[]  
list2=[]  
  
# All numbers in list1 are in form: 4*i-3  
for I in range(1,21):
```



```
list1.append(4*i-3)
```

```
# list2 stores info of odd numbers in list1
```

```
for I in range(0,20):
```

```
list2.append(list1[i]%2==1)
```

```
print('See whether all numbers in list1 are odd =>')
```

```
print(all(list2))
```

```
See whether all numbers in list1 are odd =>
```

```
True
```

פעולות לוגיות מורכבות.

בשפת פייתון קיימים 3 פונקציות לחיבור תנאים לוגיים: or, and, not.

טבלאות אמת של פונקציות אלה:

A	not A
0	1
1	0

A	B	A and B	A or B
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	1

```
>>>y=10
```

```
>>>x=5
```

```
>>>y<15 and x>8
```

```
False
```

```
>>>not y<15
```

```
False
```

```
>>>a=5
```

```
>>>b=0
```

```
>>>not a
```

```
False
```

```
>>>not b
```

```
True
```

הגדר שני משתנים שלמים עם ערכים שונים.

בעזרת אופרטור and ושני משתנים שהגדרת תבנה מערכת שתחזיר פעם אחד ערך True ופעם

שנייה ערך False.

בעזרת אופרטור or ושני משתנים שהגדרת תבנה מערכת שתחזיר פעם אחד ערך True ופעם שנייה

ערך False.

משתנים ומספרים שלמים (int).  
פעולות בסיסיות עם הסבר.

פעולה:	הסבר:
x+y	חיבור
x-y	חיסור
x*y	כפל
x/y	חילוק
x//y	חלק שלם של חילוק
x%y	שארית של חילוק
-x	הפיכת סימן
abs(x)	ערך מוחלט של מספר
divmod(x,y)	גם חלק שלם לאחר חילוק וגם שארית
x**y	חזקה
pow(x,y,[z])	גם כן חזקה, פרמטר Z הוא אופציונלי – נוהג שארית של חלוקה – זאת אומרת X בחזקה Y, שארית חלוקה ב-Z.
max(x1,x2...xn)	ערך מקסימלי בין סדרת הערכים
sum(x1,x2...xn)	סכום של סדרת הערכים
min(x1,x2...xn)	ערך מינימלי בין סדרת הערכים

```
>>> 255 + 34
289
>>> 5 * 2
10
>>> 20 / 3
6.666666666666667
>>> 20 // 3
6
>>> divmod(20,7)
(2, 6)
>>> 20 % 3
2
>>> 3 ** 4
81
>>> pow(3, 4)
81
>>> pow(3, 4, 27)
0
>>> a=(10,20,30,40,30,20,10)
>>> min(a)
10
>>> max(a)
40
>>> sum(a)
160
```

## פעולות על סיביות בעזרת מספרים (משתנים) שלמים.

פעולה:	הסבר:
$x y$	פעולת OR
$x^y$	פעולת XOR
$x&y$	פעולת AND
$x<<n$	הזזה שמאלה
$x>>y$	הזזה ימינה
$\sim x$	פעולת NOT

## הצגת מספר בבסיסים שונים:

פעולה:	הסבר:
<code>int(x,[base])</code>	הפיכת מחרוזת מבסיס כלשהו מ-2 עד 36 כולל לבסיס 10. במידה ולא צוין בסיס, אז באופן אוטומטי זה אומר שמספר הוא בבסיס 10.
<code>bin(x)</code>	הפיכת מספר לבסיס 2
<code>hex(x)</code>	הפיכת מספר לבסיס 16
<code>oct(x)</code>	הפיכת מספר לבסיס 8

```
>>> a = int('19') # העברת מחרוזת למספר
>>> b = int('19.5') # אפשר להעביר מחרוזת שמייצגת מספר עם נקודה עשרונית- אי
Traceback (most recent call last):
  File "", line 1, in
ValueError: invalid literal for int() with base 10: '19.5'
>>> c = int(19.5) # מספר ממשי נהפך למספר שלם
>>> print(a, c)
19 19
>>> bin(19)
'0b10011'
>>> oct(19)
'0o23'
>>> hex(19)
'0x13'
>>> 0b10011 # ניתן לכתוב מספר בבינרי או באוקטלי או בהקסה דצימלי
19
>>> int('10011', 2)
19
>>> int('0b10011', 2)
19
```

פונקציה abs()  
מחזירה ערך מוחלט של המספר.

```
>>> abs(-12.34)
12.34
>>> abs(1234)
1234
```

פונקציה round()

מעגלת את המספר למספר שלם קרוב ביותר. במידה ויש פרמטר נוסף, הוא מגדיר כמה נקודות אחרי נקודה צריכים ליישר. יכול להיות שפרמטר הזה יהיה שלילי – זה אומר שאנו מעגלים אחדות עשרות, מאות וכו'.

```
>>> a=10/3
>>> a
3.3333333333333335
>>> round(a,6)
3.333333
>>> round(a)
3
>>> round(a,3)
3.333
>>> round(a,-1)
0.0
>>> round(12341234.1234)
12341234
>>> round(12341234.1234,-1)
12341230.0
>>> round(12341234.1234,-2)
12341200.0
>>> print("Number:", round(a,2))
Number: 3.33
```

פונקציה int(x) מחזירה רק חלק שלם של מספר מסוג float ללא חלק העשרוני.

משתנים מרוכבים ופעולות עם מספרים מרוכבים, (complex):  
 פונקציות בסיסיות לטיפול במספרים מרוכבים.  
 שימו לב שאין לכתוב  $a=j$ , אלא צריכים לכתוב  $a=1j$ , כי  $j$  יכול להיות שם של משתנה.

אופרטור:	משמעות:
+	חיבור
-	חיסור
*	כפל
/	חילוק
<code>x=complex(a,b)</code>	הגדרת מספר מרוכב משני מספרים לא מרוכבים, במקום ראשון ממשי, שני - מדומה.
<code>x.conjugate()</code>	חיפוש של מספר צמוד למספר המקורי.
<code>x.imag</code>	חלק מדומה של מספר מרוכב.
<code>x.real</code>	חלק ממשי של מספר מרוכב.
<code>==</code>	שוויון בין שני מספרים
<code>!=</code>	שוני בין שני מספרים
<code>abs(a,b)</code>	ערך הווקטור של מספר מרוכב $\sqrt{a^2+b^2}$
<code>pow(num,n)</code>	חזקה כלשהי של מספר מרוכב.

שימו לב: אין להשתמש באופרטורים `<`, `>`, `<=`, `>=` בין מספרים מרוכבים.

```
>>> a=2+3j
>>> print(a)
(2+3j)
>>> x = complex(1, 2)
>>> print(x)
(1+2j)
>>> y = complex(3, 4)
>>> print(y)
(3+4j)
>>> z = x + y
>>> print(z)
(4+6j)
>>> z = x * y
>>> print(z)
(-5+10j)
>>> z = x / y
>>> print(z)
(0.44+0.08j)
>>> print(x.conjugate())
(1-2j)
>>> print(x.imag)
2.0
>>> print(x.real)
1.0
```

```

>>> print(x > y)
Traceback (most recent call last):
  File "<pyshell#15>", line 1, in <module>
    print(x > y)
TypeError: '>' not supported between instances of 'complex' and 'complex'
>>> print(x == y)
False
>>> abs(3 + 4j)
5.0
>>> pow(3 + 4j, 2)
(-7+24j)

```

### פונקציות מיוחדות של קבוצת fractions.

מודול זה מספק תמיכה לחשבון רציונלי. זה מאפשר ליצור מופע שבר מתוך מספרים שלמים, מספרים עם נקודה צפה, ספרות ומחרוזות.

ניתן לבנות צורת שבר מצמד מספרים שלמים, ממספר רציונלי אחר או ממחרוזת. יש להתייחס אליהם כלא ניתנים לשינוי.

נראה מספר דוגמאות שימוש בפונקציה הזאת:

פונקציה:	הסבר:
<pre> &gt;&gt;&gt;from fractions import Fraction &gt;&gt;&gt; print(Fraction(3,4)) 3/4 &gt;&gt;&gt; Fraction(3,4) Fraction(3, 4) &gt;&gt;&gt; print(Fraction(16,-10)) -8/5 &gt;&gt;&gt; print(Fraction()) 0 &gt;&gt;&gt; print(Fraction("3/4")) 3/4 &gt;&gt;&gt; Fraction('7e-6') Fraction(7, 1000000) &gt;&gt;&gt; print(Fraction('7e-6')) 7/1000000 &gt;&gt;&gt; Fraction(2.25) Fraction(9, 4) &gt;&gt;&gt; print(Fraction(2.25)) 9/4 </pre>	<p>הפעלת ספרייה בשם Fraction פעולת שבר בין שני מספרים שלמים</p> <p>פעולת שבר בין שני מספרים שלמים</p> <p>פעולת שבר בין שני מספרים שלמים עם צמצום</p> <p>פונקציה ללא פרמטרים עם ערכים של ברירת מחדל תחזיר שבר עם 0 במונה, ו 1 במכנה. הפונקציה מסוגלת לתרגם מחרוזת לשבר</p> <p>פונקציה שמקבלת רק פרמטר אחד מציבה אותו במונה ומציבה אוטומטית 1 במכנה.</p> <p>פונקציה מסוגלת לעבוד עם מספרים לא שלמים.</p>

במידה ורוצים להגביל את כמות הספרות יש להשתמש בפונקציה בשם `.limit_denominator()`.  
לדוגמה:

```
>>> Fraction('3.1415926535897932').limit_denominator(1000)
Fraction(355, 113)
>>> Fraction('3.1415926535897932').limit_denominator(1000000)
Fraction(3126535, 995207)
```

דוגמה בשימוש של פונקציות:

```
from fractions import Fraction
print(Fraction(113, 100) + Fraction(25, 18))
# returns Fraction(2267, 900)
print(Fraction(18, 5) / Fraction(18, 10))
# returns Fraction(2, 1)
print(Fraction(18, 5) * Fraction(16, 19))
# returns Fraction(288, 95)
print(Fraction(18, 5) * Fraction(15, 36))
# returns Fraction(3, 2)
print(Fraction(12, 5) ** Fraction(12, 10))
# returns 2.8592589556010197
```

```
2267/900
2
288/95
3/2
2.8592589556010197
```

תרגיל:

הגדירו משתנה INT בערך 10, משתנה FLOAT בערך 8.4, משתנה STR בערך "No".  
תכפילו את הערך של INT ב-3.5 ותעתיקו ערך הזה למשתנה BIG.  
תחסירו את הערך שנמצא במשתנה FLOAT ב-1.  
תדפיסו מנות החילוק של INT ב-FLOAT ו-BIG ב-FLOAT.  
בעזרת פעולות + ו- \* תשנה את הערך שנמצא במשתנה STR לערך "NoYesNoYesNoYes".  
תדפיסו את כל המשתנים שהשתמשתם בתוכנה הזאת.  
\*\*\* תשמרו את כל המשתנים בקובץ `myfirst.txt`

## פונקציית קליטת נתונים input.

נראה כמה דוגמאות של שימוש בפונקציה input – זאת פונקציה הקולטת מהמשתמש נתונים שונים. עקב כך שפונקציה הזאת קולטת אך ורק מחרוזות, יכול להיות מצב לא נעים שהתוצאה תוצג באופן שלא התכוונו אליו.

```
>>> input()
Yes!!!!!!
'Yes!!!!!!'
>>> answr=input()
Kuku
>>> answr
'Kuku'
>>> print(answr)
Kuku
>>> a=input()
3
>>> b=input()
4
>>> c=a+b
>>> print(c)
34
```

בואו ונראה דוגמה של סקריפט:

```
nameUser=input("What is Your name?")
cityUser=input("Where are You from?")
print("Your name is {0}. You are from {1}.".format(nameUser, cityUser))
```

```
C:\Users\marka\AppData\Local\Programs\Python\Python38-32\python.exe C:/MyPython/third.py
What is Your name? Mark
Where are You from?Karmiel
Your name is Mark. You are from Karmiel.
```

## הפיכת מחרוזת למספר.

כדי להפוך מחרוזת שקלטנו למספרים יש להשתמש בפקודות שינוי סוגי הנתונים שלמדנו בפרק הקודם.

```
qtyOranges = int(input("How many oranges? "))
priceOrange = float(input("What is the price? "))
sumOranges = qtyOranges * priceOrange
print("You had to pay", sumOranges, "shekel.")
```



```
C:\Users\marka\AppData\Local\Programs\Python\Python38-32\python.exe C:/MyPython/third.py
How many oranges? 5
What is the price? 1.12
You had to pay 5.6000000000000005 shekel.
```

תרגילים:

כתבו תוכנית הקולטת מהמשתמש: שם, שם משפחה, שנת לידה ומדפיסה את הנתונים האלה, ולאחר מכן קולטת מהמשתמש שנה נוכחית ומחשבת בן כמה המשתמש.

כתבו תוכנית שמדפיסה על מסך תרגיל: 4\*100-54 ומבקשת מהמשתמש להכניס את התשובה. אחרי זה היא מדפיסה את התוצאה הנכונה ואת התוצאה שמשתמש הכניס.

כתבו תוכנית בה תבקשו מהמשתמש להכניס 4 מספרים. תחברו מספר ראשון עם מספר שני ומספר שלישי עם מספר רביעי. תחלקו סכום ראשון בסכום השני ותדפיסו את התשובה על המסך עם 2 ספרות אחרי נקודה.

למרות כל הגמישות של שפת פייתון אין אפשרות לחבר מחרוזות עם מספר. כדי לאפשר חיבור של סוגים שונים צריך להמיר מסוג אחד לסוג שני. פונקציה `str()` מעבירה משתנה למחרוזת, `int()` למספר שלם, `float()` למספר ממשי, `complex()` למספר מדומה.

כמובן שכדי להפוך נתון מסוג אחד לסוג צריך להיות מכנה משותף. אי אפשר להפוך "I" למספר ו-"3.2j" לשלם. אבל מספרים יכולים להיות מחרוזות, או שמספר שהוא כבר מחרוזת יכול להפוך למספר חזרה

```
>>> 1+'a'
Traceback (most recent call last):
  File "<pyshell#0>", line 1, in <module>
    1+'a'
TypeError: unsupported operand type(s) for +: 'int' and 'str'
>>> str(1)+'a'
'1a'
>>> int('3')+4
7
>>> float('3.2')+int('2')
5.2
>>> str(4)+str(1.2)
'41.2'
>>> 3+complex("5-2j")
(8-2j)
>>> int('a')
Traceback (most recent call last):
  File "<pyshell#5>", line 1, in <module>
    int('a')
ValueError: invalid literal for int() with base 10: 'a'
>>> int(2j)
Traceback (most recent call last):
  File "<pyshell#8>", line 1, in <module>
    int(2j)
TypeError: can't convert complex to int
```

לפונקציה `int()` יש תפקיד מיוחד – בעזרתה יש אפשרות להעביר בסיסיים, מבסיס אחד לחזרה לבסיס 10. שימו לב שהמספר בפנים צריך להיות בצורת מחרוזת, והמספר לידו הוא הבסיס ממנו נרצה להמיר את המספר

```
>>> int('1001',2)
9
```

```
>>> int('F',16)
15
>>> int(723,8)
Traceback (most recent call last):
  File "<pyshell#11>", line 1, in <module>
    int(723,8)
TypeError: int() can't convert non-string with explicit base
>>> int('723',8)
467
>>> int('723',9)
588
>>> int('123210',5)
4805
```

## גנרטור של מספר אקראי:

בתוכנות מחשב נדרש לעתים קרובות לדמות אקראיות, לדוגמה בעת פיתוח משחקים. אם לתוכנית יש גנרטור מסוים, כלומר מפיק של מספר אקראי, ואז באמצעות המספר המתקבל בצורה זו, תוכלו לבחור ענף כזה או אחר של ביצוע התוכנית, או אובייקט שרירותי מהאוסף. במילים אחרות, העיקר לייצר את המספר. סוג אחר של אמולציה אקראית מבוסס עליו.

אנחנו כנראה לא יודעים אם יש סיכוי בטבע, או שזה נראה לנו רק בגלל מגבלות הידע שלנו. אנחנו רק יודעים שבתכנות אין צירוף מקרים אמיתי. אין לנו אפשרות לקחת מספר שרירותי, לא ניתן לתכנת את הופעתו משום מקום. ניתן רק ליצור תוכנית שכתוצאה מיישום נוסחה מורכבת על "אירוע" ליצירת מספר, ולנו כמשתמשים פשוטים נראה שהמספר הזה הוא אקראי.

"אירוע" הוא הקלט לפורמולה. זה יכול להיות למשל, זמן המערכת באלפיות השנייה, שמשנתה כל הזמן. כתוצאה מכך, תוצאת התרחשות "האירוע" תהיה כל הזמן שונה. או שהמתכנת יכול להגדיר זאת לבד.

תוכנית דומה (במציאות, מודול או פונקציה) נקראת מחולל מספרים פסאודו (כאילו) אקראיים. ספריית פייתון הסטנדרטית כוללת מודול אקראי. הוא מכיל פונקציות רבות הקשורות למספרים אקראיים, למשל "ערבוב" של אלמנטים ברצף ולא רק פונקציות של מספרים פסאודו אקראיים.

שימו לב כי המודול random מכיל את הפונקציה random() של אותו שם.

כדי לגשת לפונקציות, יש לייבא את המודול האקראי:

```
>>> import random
```

או:

```
>>> from random import random, randrange, randint
```

נבדוק קודם כל פונקציות לקבלת מספרים אקראיים שלמים: randint(), randrange().

פונקציה randint() מקבלת גבולות של שני מספרים ומחזירה מספר אקראי בין גבולות האלה (כולל קצבות [a; b]). מספרים יכולים להיות גם חיובים וגם שליליים. שימו לב שמספר קטן צריך להיות במקום ראשון ומספר גדול צריך להיות במקום שני, אחרת תקבלו שגיאת קומפילציה. אין בעיה ששני מספרים יהיו זהים, אך אין הגיון בזה.

```
>>> import random
>>> random.randint(0,10)
9
>>> random.randint(-100,-10)
-66
>>> random.randint(-100,100)
-44
>>> random.randint(0,1)
1
```

```

>>> random.randint(-1,1)
-1
>>> random.randint(10,-10)
Traceback (most recent call last):
  File "<pyshell#6>", line 1, in <module>
    random.randint(10,-10)
  File "C:\Users\marka\AppData\Local\Programs\Python\Python38-32\lib\random.py", line 248, in
  randint
    return self.randrange(a, b+1)
  File "C:\Users\marka\AppData\Local\Programs\Python\Python38-32\lib\random.py", line 226, in
  randrange
    raise ValueError("empty range for randrange() (%d, %d, %d)" % (istart, istop, width))
ValueError: empty range for randrange() (10, -9, -19)
>>> random.randint(0,0)
0

```

פונקציה `randrange()` היא יותר מורכבת – היא יכולה לקבל פרמטר אחד, שני פרמטרים או שלושה פרמטרים.

במידה וצוין פרמטר אחד פונקציה מחזירה ערך בין 0 עד פרמטר (לא כולל אותות) –  $[0,a)$ .

אם מציינים שני פרמטרים, פונקציה מחזירה ערך אקראי בין הפרמטרים כולל גבול תחתון ולא כולל גבול עליון –  $[a,b)$ .

במידה ונותנים 3 פרמטרים, פרמטר שלישי מציין צעד של "זריקת" מספר. במידה ונרשום גבולות בין 20 עד 100 ופרמטר השלישי יהיה 2 כל המספרים שנקבל יהיו זוגיים.

```

>>> random.randrange(100)
32
>>> random.randrange(-2)
Traceback (most recent call last):
  File "<pyshell#9>", line 1, in <module>
    random.randrange(-2)
  File "C:\Users\marka\AppData\Local\Programs\Python\Python38-32\lib\random.py", line 216, in
  randrange
    raise ValueError("empty range for randrange()")
ValueError: empty range for randrange()

```

```

>>> random.randrange(-10,10)
4
>>> random.randrange(10,2)
Traceback (most recent call last):
  File "<pyshell#11>", line 1, in <module>
    random.randrange(10,2)

```

```

File "C:\Users\marka\AppData\Local\Programs\Python\Python38-32\lib\random.py", line 226, in
randrange
    raise ValueError("empty range for randrange() (%d, %d, %d)" % (istart, istop, width))
ValueError: empty range for randrange() (10, 2, -8)
>>> random.randrange(1,1)
Traceback (most recent call last):
  File "<pyshell#12>", line 1, in <module>
    random.randrange(1,1)
  File "C:\Users\marka\AppData\Local\Programs\Python\Python38-32\lib\random.py", line 226, in
randrange
    raise ValueError("empty range for randrange() (%d, %d, %d)" % (istart, istop, width))
ValueError: empty range for randrange() (1, 1, 0)
>>> random.randrange(0,1)
0

```

```

>>> random.randrange(0,100,5)
70
>>> random.randrange(0,100,5)
15
>>> random.randrange(0,100,5)
30
>>> random.randrange(0,100,5)
85
>>> random.randrange(0,100,5)
90
>>> random.randrange(0,100,5)
40
>>> random.randrange(0,100,5)
45

```

פונקציה random() מחזירה מספר אקראי בין 0 לבין 1 (לא כולל 1). אם נרצה לעגל אותו – ניתן להשתמש בפונקציה round().

```

>>> a=random.random()
>>> a
0.3575734730649053
>>> round(a,2)
0.36
>>> round(random.random(),5)
0.84074

```

נניח שאנו רוצים לקבל מספר בין 0 לבין 10.

(באופן כללי יש להכפיל בגבול עליון).

```
>>> random.random() * 10
0.8378949783739364
>>> random.random() * 10
7.000003967417433
>>> random.random() * 10
8.678314956061422
>>> random.random() * 10
2.35202306391755
```

נניח שאנו רוצים לקבל מספר בין 4 לבין 10.

(באופן כללי יש להכפיל בהפרש בין תחום עליון לגבול תחתון ולהוסיף גבול תחתון).

```
>>> random.random() * (10 - 4) + 4
6.734976502483406
>>> random.random() * (10 - 4) + 4
5.754145287863961
>>> random.random() * (10 - 4) + 4
7.780705378014292
>>> random.random() * (10 - 4) + 4
4.294794183653732
>>> random.random() * (10 - 4) + 4
4.307891310704934
```

דוגמה של מספר אקראי בין -1 לבין 1.

```
>>> random.random() * (1 + 1) - 1
0.47626619668277126
>>> random.random() * (1 + 1) - 1
-0.9358616346304338
>>> random.random() * (1 + 1) - 1
0.2754603241754101
>>> random.random() * (1 + 1) - 1
-0.4101838883634219
```

האמת שתמיד אפשר להשתמש רק בפונקציה `random()`:

מספר אקראי בין 0 לבין 100 (לא כולל 100).

```
>>> int(random.random() * 100)
24
>>> int(random.random() * 100)
56
>>> int(random.random() * 100)
```

מספר אקראי בין 50- לבין 50 (לא כולל 50).

```
>>> round(random.random() * 100 - 50)
-34
>>> round(random.random() * 100 - 50)
-28
>>> round(random.random() * 100 - 50)
44
>>> round(random.random() * 100 - 50)
24
```

## קריאה ושימוש במודולים.

### כללים לגבי שימוש במודולים:

מי שלמד שפת C מכיר מצויין את השורה - `#include "stdio.h"` – זאת קריאה לספריה סטנדרטית של קלט ופלט. בשפת פייתון לספריות יש שם "מודול". הפונקציות המובנות בשפת התכנות זמינות מידית כדי לקרוא להם, אין צורך לבצע קריאה למודול. עם זאת, במהלך קיומה של כל שפה פופולרית נכתבו עליה כל כך הרבה פונקציות ומחלקות, שנמצאו מבוקשים על ידי מתכנתים רבים בתחומים שונים, שזה לא מעשי לכלול את כל הקוד הזה בשפה עצמה. כדי לפתור את בעיית הגישה לתכונות נוספות של השפה, זה הפך למקובל בתכנות להשתמש במודולים, חבילות וספריות. כל מודול מכיל אוסף של פונקציות ומחלקות שנועדו לפתור בעיות מתחום ספציפי. כך שמודול `math` של פייתון מכיל פונקציות מתמטיות, המודול `random` מאפשר ליצור מספרים אקראיים, מודול `datetime` מכיל מחלקות לעבודה עם תאריכים ושעות, מודול ה-`sys` מספק גישה למשתני מערכת וכו'. מספר המודולים של פייתון הוא עצום, עקב הפופולריות של השפה. חלק מהמודולים נאספים בספריות שנקראת סטנדרטיות. סטנדרטי מכיוון שהוא מגיע עם חבילת ההתקנה. עם זאת, ישנן ספריות של צד שלישי, מורידים ומתקינים אותם בנפרד. כדי לגשת לפונקציונליות המודול, יש לייבא אותה לתוכנית. לאחר הייבוא, המתורגמן "יודע" על קיומם של מחלקות ופונקציות נוספות ומאפשר להשתמש בהן. בפיייתון יבוא מתבצע באמצעות פקודת `import`. ישנן כמה דרכים לייבא. נראה את הדוגמה של עבודה עם מודול מתמטיקה.

```
>>> import math
>>>
```

תוכנה לא מחזירה ולא כלום, אבל מאחורי התוכנית מודול זה מצורף ומוכן. אם נכתוב `math` נראה:

```
>>> math
<module 'math' (built-in)>
```



כדי לראות את כל הפקודות ששייכות למודול זה נכתוב `dir(math)` ונראה את רשימת הפקודות:

```
>>> dir(math)
['__doc__', '__loader__', '__name__', '__package__', '__spec__', 'acos', 'acosh', 'asin', 'asinh',
'atan', 'atan2', 'atanh', 'ceil', 'comb', 'copysign', 'cos', 'cosh', 'degrees', 'dist', 'e', 'erf', 'erfc', 'exp',
'expm1', 'fabs', 'factorial', 'floor', 'fmod', 'frexp', 'fsum', 'gamma', 'gcd', 'hypot', 'inf', 'isclose',
'isfinite', 'isinf', 'isnan', 'isqrt', 'ldexp', 'lgamma', 'log', 'log10', 'log1p', 'log2', 'modf', 'nan', 'perm',
'pi', 'pow', 'prod', 'radians', 'remainder', 'sin', 'sinh', 'sqrt', 'tan', 'tanh', 'tau', 'trunc']
```

פקודות כמו: `'__doc__'` משמשות את המודול לפעולות פנימיות שלו, ולא משמשות את המתכנת כלל. כל השאר הם שמות הפונקציות וקבועים (משתנים שאינם משנים את ערכיהם), הכלולים במודול המתמטיקה.

כדי להפעיל פונקציה ממודול, כתוב את שם המודול מקדימה, שים נקודה, ציין את שם הפונקציה ואז העביר את הארגומנטים בסוגריים אם הם נדרשים.

לדוגמה, כדי לקרוא לפונקציית `pow`, היית כותב את זה:

```
math.pow()
```

שימו לב שפונקציה `math.pow` שונה מפונקציה מובנת `pow`.

```
>>> pow(2,5)
32
>>> math.pow(2,5)
32.0
```

כדי לקבל עזרה על פקודה מסוימת נכתוב:

```
>>> help(math.gcd)
Help on built-in function gcd in module math:

gcd(x, y, /)
  greatest common divisor of x and y

>>> math.gcd(81,27)
27
```

כדי לראות ערך של קבוע, מספיק לכתוב שם של קבוע לאחר ".":

```
>>> math.pi
3.141592653589793
>>> math.e
2.718281828459045
```

אפשרות שנייה – להביא פונקציות נדרשות ישירות לתוכנה:

או חלק מהפונקציות.

```
>>> from math import gcd, sqrt, pi
>>> pi
3.141592653589793
>>> sqrt(9)
3.0
>>> gcd(64,80)
16
>>> sin(1)
Traceback (most recent call last):
  File "<pyshell#75>", line 1, in <module>
    sin(1)
NameError: name 'sin' is not defined
```

או את כולם.

```
>>> from math import *
>>> sin(1)
0.8414709848078965
```

במקרה הזה אין צורך לכתוב `.math.gcd(81,27)`.  
ניתן לקשר קבועים למשתנים אחרים:

```
>>> from math import pi as P
>>> P
3.141592653589793
```

דוגמה אחרת:

```
>>> import calendar
>>> calendar.weekheader(2)
'Mo Tu We Th Fr Sa Su'
```

או:

```
>>> from calendar import weekheader as week
>>> week(3)
'Mon Tue Wed Thu Fri Sat Sun'
```

כל מתכנת יכול לבנות מודול משלו. נעשה דוגמה של מודול לחישוב שטחים:

```
from math import pi, pow
def rectangle(a, b):
    return round(a * b, 2)
def triangle(a, h):
```

```

return round(0.5 * a * h, 2)
def circle(r):
return round(pi * pow(r, 2), 2)

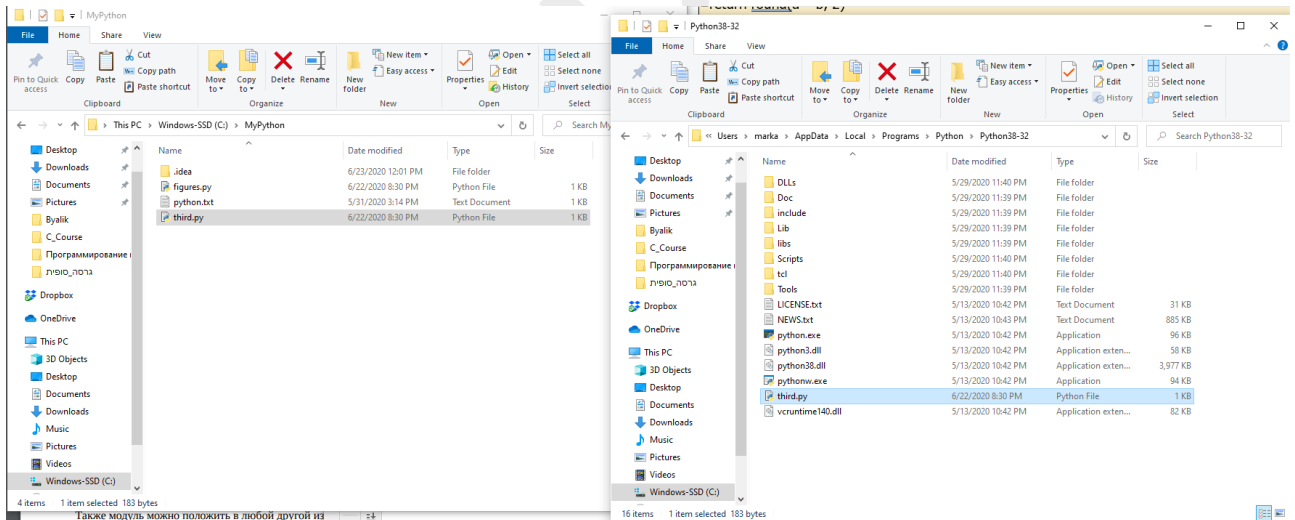
```

מודול שכתבנו נשמור בקובץ figures.py. צריכים לשמור את הקובץ במקום, כך שפייתון יכול לראות אותו. אפשר לבדוק את מיקום של ספריות שמוכרות ע"י פייתון. נעשה את זה ע"י מודול sys. במידה ונשים קובץ לאחד מהמקומות האלו פייתון יוכל למצוא אותם.

```

>> import sys
>>> sys.path
['', 'C:\\Users\\marka\\AppData\\Local\\Programs\\Python\\Python38-32\\Lib\\idlelib',
'C:\\Users\\marka\\AppData\\Local\\Programs\\Python\\Python38-32\\python38.zip',
'C:\\Users\\marka\\AppData\\Local\\Programs\\Python\\Python38-32\\DLLs',
'C:\\Users\\marka\\AppData\\Local\\Programs\\Python\\Python38-32\\lib',
'C:\\Users\\marka\\AppData\\Local\\Programs\\Python\\Python38-32',
'C:\\Users\\marka\\AppData\\Local\\Programs\\Python\\Python38-32\\lib\\site-packages']

```



נחשב שטח של מלבן, משולש ומעגל:

```

import figures
a=figures.rectangle(2,3)
print(a)
b=figures.triangle(6,4)
print(b)
c=figures.circle(5)
print(c)

```

6  
12.0  
78.54

מודול של מספרים מרוכבים – מודול cmath.

הסבר	פונקציה
פונקציה מחזירה זווית של מספר מרוכב בטווח בין $[-\pi, \pi]$ דומה למשפט הבא: $\text{math.atan2}(x.\text{imag}, x.\text{real})$	$\text{cmath.phase}(x)$
המרה של מספר מצורה אלגברית לצורה פולארית – פונקציה מחזירה $(r, \text{phi})$	$\text{cmath.polar}(x)$
המרה של מספר מצורה פולארית לצורה אלגברית.	$\text{cmath.rect}(r, \text{phi})$
פונקציה $e^x$	$\text{cmath.exp}(x)$
פונקציה מחזירה לוגריתם לפי בסיס $\text{base}$ , במידה ובסיס לא צוין פונקציה מחזירה לוגריתם לפי בסיס $e$ .	$\text{cmath.log}(x[, \text{base}])$
פונקציה מחזירה לוגריתם לפי בסיס 10	$\text{cmath.log10}(x)$
פונקציה מחזירה שורש ריבועי	$\text{cmath.sqrt}(x)$
פונקציה מחזירה ארק קוסינוס של מספר (זווית בין $[-\pi, \pi]$ )	$\text{cmath.acos}(x)$
פונקציה מחזירה ארק סינוס של מספר (זווית בין $[-\pi, \pi]$ )	$\text{cmath.asin}(x)$
פונקציה מחזירה ארק טנגנס של מספר (זווית בין $[-\pi, \pi]$ )	$\text{cmath.atan}(x)$
פונקציה מחזירה קוסינוס של זווית בין $[-\pi, \pi]$	$\text{cmath.cos}(x)$
פונקציה מחזירה סינוס של זווית בין $[-\pi, \pi]$	$\text{cmath.sin}(x)$
פונקציה מחזירה טנגנס של זווית בין $[-\pi, \pi]$	$\text{cmath.tan}(x)$
פונקציה מחזירה ארק קוסינוס היפרבולי של מספר (זווית בין $[-\pi, \pi]$ )	$\text{cmath.acosh}(x)$
פונקציה מחזירה ארק סינוס היפרבולי של מספר (זווית בין $[-\pi, \pi]$ )	$\text{cmath.asinh}(x)$
פונקציה מחזירה ארק טנגנס היפרבולי של מספר (זווית בין $[-\pi, \pi]$ )	$\text{cmath.atanh}(x)$
פונקציה מחזירה קוסינוס היפרבולי של זווית בין $[-\pi, \pi]$	$\text{cmath.cosh}(x)$
פונקציה מחזירה סינוס היפרבולי של זווית בין $[-\pi, \pi]$	$\text{cmath.sinh}(x)$
פונקציה מחזירה טנגנס היפרבולי של זווית בין $[-\pi, \pi]$	$\text{cmath.tanh}(x)$
פונקציה שמחזירה "אמת" במידה אם גם חלק ממשי וגם חלק מדומה סופיים.	$\text{cmath.isfinite}(x)$
פונקציה שמחזירה "אמת" במידה אם או חלק ממשי או חלק מדומה אין סופיים.	$\text{cmath.isinf}(x)$
פונקציה שמחזירה "אמת" במידה אם או חלק ממשי או חלק מדומה NaN.	$\text{cmath.isnan}(x)$
	$\text{cmath.pi}$
	$\text{cmath.e}$

תרגול בנושא פעולות בסיסיות:  
תרגול בנושא אופרטורים בסיסיים.  
תרגיל מס' 1.

מה יהיו תוצאות של כל פקודה (ללא הרצת תוכנית). תבדוק את התוצאה בעזרת Python Shell.

<code>int(9 - 3*0.3)</code>	<code>float(8 * 2.5)</code>	<code>int(9 / 2)</code>	<code>float(9 / -2)</code>
<code>9 // -2</code>	<code>9 % 2</code>	<code>9.0 % 2</code>	<code>9 % 2.0</code>
<code>9 % -2</code>	<code>-9 % 2</code>	<code>9 / -2.0</code>	<code>4 + 3 * 5</code>
<code>(4 + 3) * 5</code>			

תרגיל מס' 2.

מה יהיו תוצאות של כל פקודה (ללא הרצת תוכנית). תבדוק את התוצאה בעזרת Python Shell.

<code>6 ** 3 // 7 * 4</code>	<code>5 / 4 + 3 // 4</code>	<code>5 - 2 * 3 ** 4</code>	<code>int(34.6)</code>
<code>int(-4.3)</code>	<code>float(21)</code>	<code>round(3.8)</code>	<code>. round(3.3)</code>
<code>round(3.5)</code>	<code>round(-3.3)</code>	<code>round(-3.5)</code>	<code>11 * 2 ** 2 - 13 / 4 + 7</code>
<code>max(2, -3, min(4, 7), -5)</code>	<code>min(max(3, 4), abs(-5))</code>	<code>abs(min(4, 6, max(2, 8)))</code>	

תרגיל מס' 3.

מהן פקודות חוקיות, מה תהיה תוצאה של פקודה חוקית? תסמן אותם ללא הרצת תוכנית. תבדוק את התוצאה בעזרת Python Shell.

<code>6 * -----8</code>	<code>8 = people</code>	<code>3 + 4 / 2 - 7 % 2</code>	<code>((--((-5))))</code>
<code>4 += 7 / 2</code>	<code>operand_1 &gt;= operand_2</code>	<code>7 / (2 / (3 / 3))</code>	<code>int('4' - 2)</code>
<code>int('91.4')</code>	<code>int(524.345 ** 3534545345)</code>	<code>int('7.1 + 4')</code>	<code>int('4 - 2')</code>
<code>int('42')</code>	<code>int(-12.12)</code>	<code>print(3 * 7, (17 - 2) * 8)</code>	<code>int('123e')</code>
<code>((((4 ** 3))))</code>	<code>7 * 2 + 5 / 12 % 5</code>	<code>P = 4 * 5 // 3 % 5</code> <code>Q = 34 % P * 5 - 29 % 5 * 2</code>	<code>5 + -3</code>
<code>4 * *</code>	<code>6 * (3 + 1)</code>	<code>round(max(5.572, 3.258), abs(-2))</code>	

תרגיל מס' 4.

מה יהיו תוצאות של שגרה (ללא הרצת תוכנית). תבדוק את התוצאה בעזרת Python Shell.

<code>x = 41</code> <code>y = 41</code> <code>result = x == y</code> <code>print(result)</code>	<code>x = 41</code> <code>y = 63</code> <code>result = x == y</code> <code>print(result)</code>	<code>x = 87</code> <code>y = 63</code> <code>result = x &gt; y</code> <code>print(result)</code>	<code>x = 41</code> <code>y = 63</code> <code>result = x &gt; y</code> <code>print(result)</code>
<code>x = 41</code> <code>y = 63</code> <code>result = x &lt; y</code> <code>print(result)</code>	<code>x = 92</code> <code>y = 63</code> <code>result = x &lt; y</code> <code>print(result)</code>	<code>x = 92</code> <code>y = 63</code> <code>result = x != y</code> <code>print(result)</code>	<code>x = 63</code> <code>y = 63</code> <code>result = x != y</code> <code>print(result)</code>

x = 97 y = 63 result = x >= y print(result)	x = 42 y = 63 result = x >= y print(result)	x = 42 y = 63 result = x <= y print(result)	x = 92 y = 63 result = x <= y print(result)
x = 57 y = 12 result = x >= y print(result)	x = 8 y = 8 result = x >= y print(result)	x = 78 y = 89 result = x >= y print(result)	a = 5 b = 7 a = a + b b = a - b a = a - b

תרגיל מס' 5.

מה תהיה תוצאה של כל פעולה, כאשר נתונים n1, n2.

n1=213

n2=134

bit_or = n1   n2	b = ~n1	a = n1 >> 3
bit_and = n1 & n2	a=n2 >> 3	b=n2 << 3
bit_xor = n1 ^ n2	c=a b	c=a^b

תרגיל מס' 6.

מה היא תוצאה של הקוד הבא, כאשר קלט הוא 208, 151, 123?

```
n = int(input())
A = n % 10
B = n // 100
C = n // 10 % 10
L = (n % A == 0) & (n % B == 0) & (n % C == 0)
print(L)
```

תרגיל מס' 7.

תפענח את הקוד הבא והצג דוגמה לפלט האפשרי של הקוד.

```
import random
A = random.randrange(-10,10)
B = random.randrange(-10,10)
print("A = ", A)
print("B = ", B)
x = (A>2) and (B<=3)
print("A > 2: ", (A>2))
print("B <=3: ", (B<=3))
print("A > 2 and B <=3: ", x)
```

תרגיל מס' 8.

תפענח את הקוד הבא והצג דוגמה לפלט האפשרי של הקוד.

```
import random
A = random.randrange(-10,10)
B = random.randrange(-10,10)
print("A = ", A)
print("B = ", B)
x = (A>=0) or (B<-2)
print("A >= 0: ", (A>=0))
print("B < -2: ", (B<-2))
print("A>=0 or B<-2: ", x)
```

תרגיל מס' 9.

תפענח את הקוד הבא והצג דוגמה לפלט האפשרי של הקוד.

```
import random
A = random.randrange(-10,5)
B = random.randrange(-5,5)
C = random.randrange(-5,10)
print("A = ", A)
print("B = ", B)
print("C = ", C)
x = (A<B) and (B<C)
print("A<B: ", (A<B))
print("B<C: ", (B<C))
print("A<B<C: ", x)
```

תרגיל מס' 10.

תפענח את הקוד הבא והצג דוגמה לפלט האפשרי של הקוד.

```
import random
A = random.randrange(-10,10)
B = random.randrange(-6,6)
C = random.randrange(-10,10)
print("A = ", A)
print("B = ", B)
print("C = ", C)
x = ((A<B) and (B<C)) or ((C<B) and (B<A))
print("B is between A and C: ", x)
```

תרגיל מס' 11.

תפענח את הקוד הבא והצג דוגמה לפלט האפשרי של הקוד.

```
import random
A = random.randrange(1,10)
B = random.randrange(1,10)
print("A = ", A)
print("B = ", B)
a1 = (A%2)==1
b1 = (B%2)==1
x = a1 and b1
print("A even: ", a1)
print("B odd: ", b1)
print("A and B odd: ", x)
```

תרגיל מס' 12.

תפענח את הקוד הבא והצג דוגמה לפלט האפשרי של הקוד.

```
import random
A = random.randrange(1,10)
B = random.randrange(1,10)
print("A = ", A)
print("B = ", B)
a1 = (A%2)==1
b1 = (B%2)==1
x = a1 or b1
print("A even: ", a1)
print("B odd: ", b1)
print("or A or B odd: ", x)
```

תרגיל מס' 13.

תפענח את הקוד הבא והצג דוגמה לפלט האפשרי של הקוד.

```
import random
A = random.randrange(1,10)
B = random.randrange(1,10)
print("A = ", A)
print("B = ", B)
a1 = (A%2)==1
b1 = (B%2)==1
x = (a1 != b1)
print("A odd: ", a1)
```



```
print("B odd: ", b1)
print("only one is odd: ", x)
```

תרגיל מס' 14.

תפענח את הקוד הבא והצג דוגמה לפלט האפשרי של הקוד.

```
import random
A = random.randrange(1,10)
B = random.randrange(1,10)
print("A = ", A)
print("B = ", B)
a1 = (A%2)==0
b1 = (B%2)==0
x = (a1 == b1)
print("A even: ", a1)
print("B even: ", b1)
print("A and B even or odd: ", x)
```

תרגיל מס' 15.

תפענח את הקוד הבא והצג דוגמה לפלט האפשרי של הקוד.

```
import random
A = random.randrange(-10,10)
B = random.randrange(-10,10)
C = random.randrange(-10,10)
print("A = ", A)
print("B = ", B)
print("C = ", C)
a = (A>0)
b = (B>0)
c = (C>0)
x = (a and b and c)
print("A positive: ", a)
print("B positive: ", b)
print("C positive: ", c)
print("Each of A, B, C positive: ", x)
```

תרגיל מס' 16.

תפענח את הקוד הבא והצג דוגמה לפלט האפשרי של הקוד.

```
import random
A = random.randrange(-10,10)
B = random.randrange(-10,10)
C = random.randrange(-10,10)
print("A = ", A)
print("B = ", B)
print("C = ", C)
a = (A>0)
b = (B>0)
c = (C>0)
x = (a or b or c)
print("A positive: ", a)
print("B positive: ", b)
print("C positive: ", c)
print("At least one of A, B, C positive: ", x)
```

תרגיל מס' 17.

תפענח את הקוד הבא והצג דוגמה לפלט האפשרי של הקוד.

```
import random
A = random.randrange(-10,10)
B = random.randrange(-10,10)
C = random.randrange(-10,10)
print("A = ", A)
print("B = ", B)
print("C = ", C)
a = (A>0)
b = (B>0)
c = (C>0)
x = (a and not b and not c) or (not a and b and not c) or (not a and not b and c)
print("A positive: ", a)
print("B positive: ", b)
print("C positive: ", c)
print("Only one of A, B, C is positive: ", x)
```

תרגיל מס' 18.

תפענח את הקוד הבא והצג דוגמה לפלט האפשרי של הקוד.

```
import random
A = random.randrange(-10,10)
B = random.randrange(-10,10)
C = random.randrange(-10,10)
print("A = ", A)
print("B = ", B)
print("C = ", C)
a = (A>0)
b = (B>0)
c = (C>0)
x = (a and b and not c) or (not a and b and c) or (a and not b and c)
print("A positive: ", a)
print("B positive: ", b)
print("C positive: ", c)
print("Only two from A, B, C are positive: ", x)
```

תרגיל מס' 19.

תפענח את הקוד הבא והצג דוגמה לפלט האפשרי של הקוד.

```
import random
A = random.randrange(1,200)
print("A = ", A)
a = (A%2 == 0)
b = (A>=10 and A<=99)
x = (a and b)
print("A even: ", a)
print("A two cyphers: ", b)
print("The number is even and two cyphers: ", x)
```

תרגיל מס' 20.

תפענח את הקוד הבא והצג דוגמה לפלט האפשרי של הקוד.

```
import random
A = random.randrange(1,200)
print("A = ", A)
a = (A%2 == 1)
b = (A>=100 and A<=999)
x = (a and b)
print("A odd: ", a)
```

```
print("A 3 cyphers: ", b)
print("This number odd and 3 cyphers: ", x)
```

תרגיל מס' 21.

תפענח את הקוד הבא והצג דוגמה לפלט האפשרי של הקוד.

```
import random
A = random.randrange(-10,10)
B = random.randrange(-10,10)
C = random.randrange(-10,10)
print("A = ", A)
print("B = ", B)
print("C = ", C)
ab = (A == B)
bc = (B == C)
ac = (A == C)
x = ab or bc or ac
print("A equal B: ", ab)
print("B equal C: ", bc)
print("A equal C: ", ac)
print("At least one pair is equal ", x)
```

תרגיל מס' 22.

תפענח את הקוד הבא והצג דוגמה לפלט האפשרי של הקוד.

```
import random
A = random.randrange(-3,3)
B = random.randrange(-3,3)
C = random.randrange(-3,3)
print("A = ", A)
print("B = ", B)
print("C = ", C)
ab = (A == -B)
bc = (B == -C)
ac = (A == -C)
x = ab or bc or ac
print("A is opposite B: ", ab)
print("B is opposite C: ", bc)
print("A is opposite C: ", ac)
print("At least one pair of opposite pairs: ", x)
```

תרגיל מס' 23.

תפענח את הקוד הבא והצג דוגמה לפלט האפשרי של הקוד.

```
import random
N = random.randrange(100,1000)
print("Number: ", N)
d2 = int(N/100)
d1 = int((N-d2*100)/10)
d0 = N%10
print("Hundreds: ", d2)
print("Tens: ", d1)
print("Ones: ", d0)
ab = (d2 != d1)
bc = (d1 != d0)
ac = (d2 != d0)
x = ab and bc and ac
print("All digits are different: ", x)
```

תרגיל מס' 24.

תפענח את הקוד הבא והצג דוגמה לפלט האפשרי של הקוד.

```
import random
N = random.randrange(100,1000)
print("Number: ", N)
d2 = int(N/100)
d1 = int((N-d2*100)/10)
d0 = N%10
print("Hundreds: ", d2)
print("Tens: ", d1)
print("Ones: ", d0)
x = (d2 < d1) and (d1 < d0)
print("Cyphers are increasing: ", x)
```

תרגיל מס' 25.

תפענח את הקוד הבא והצג דוגמה לפלט האפשרי של הקוד.

```
import random
N = random.randrange(100,1000)
N = 379
print("Number: ", N)
d2 = int(N/100)
d1 = int((N-d2*100)/10)
```

```

d0 = N%10
print("Hundreds: ", d2)
print("Tens: ", d1)
print("Ones: ", d0)
x = ((d2 < d1) and (d1 < d0)) or ((d2 > d1) and (d1 > d0))
print("Cyphers are increasing or decreasing", x)

```

תרגיל מס' 26.

תפענח את הקוד הבא והצג דוגמה לפלט האפשרי של הקוד.

```

import random
N = random.randrange(1000,10000)
print("Number: ", N)
p1 = int(N/100)
p2 = N-p1*100
p2_d1 = int(p2/10)
p2_d0 = p2 - p2_d1*10
print("First half 1: ", p1)
print("Second half 2: ", p2)
print("Tens: ", p2_d1)
print("Ones: ", p2_d0)
x = (p1 == (p2_d0*10 + p2_d1))
print("Palindrome: ", x)

```

תרגיל מס' 27.

תפענח את הקוד הבא והצג דוגמה לפלט האפשרי של הקוד.

```

import random
A = random.randrange(-10,10)
B = random.randrange(-10,10)
C = random.randrange(-10,10)
print("A = {0}, B = {1}, C = {2}".format(A,B,C))
D = B*B - 4*A*C
x = (D >= 0)
print("D=:",D)
print("Equation {0}x^2 + {1}x + {2} = 0 has real roots: ".format(A,B,C), x)

```

תפענח את הקוד הבא והצג דוגמה לפלט האפשרי של הקוד.

```
import random
x = random.randrange(-10,10)
y = random.randrange(-10,10)
print("x = {0}, y = {1}".format(x,y))
b = (x > 0 and y > 0)
print("Point is in the first quarter: ", b)
b = (x < 0 and y > 0)
print("Point is in the second quarter: ", b)
b = (x < 0 and y < 0)
print("Point is in the third quarter: ", b)
b = (x > 0 and y < 0)
print("Point is in the fourth quarter: ", b)
b=x==0
print("Point is on X axis: ", b)
print("Point is in the fourth quarter: ", b)
b=y==0
print("Point is on Y axis: ", b)
```

תפענח את הקוד הבא והצג דוגמה לפלט האפשרי של הקוד.

```
import random
x1,x2 = sorted(random.sample(range(-10, 11), 2))
y2,y1 = sorted(random.sample(range(-10, 11), 2))
x = random.randrange(-10,11)
y = random.randrange(-10,11)
b = (x1 < x) and (x < x2) and (y2 < y) and (y < y1)
print("Point (x1, y1): ({0},{1})".format(x1, y1))
print("Point (x2, y2): ({0},{1})".format(x2, y2))
print("Point (x, y): ({0},{1})".format(x, y))
print("Point inside the rectangle: ",b)
```

תרגיל מס' 30.

תפענח את הקוד הבא והצג דוגמה לפלט האפשרי של הקוד.

```
import random
N = random.randrange(100,1000)
print("Num: ", N)
d2 = int(N/100)
d1 = int((N-d2*100)/10)
d0 = N%10
print("Hundreds",d2)
print("Tens: ", d1)
print("Ones: ", d0)
```

תרגיל מס' 31.

תפענח את הקוד הבא והצג דוגמה לפלט האפשרי של הקוד.

```
import random
N = random.randrange(100,1000)
print("Num: ", N)
d2 = int(N/100)
d1 = int((N-d2*100)/10)
d0 = N%10
print("Hundreds: ", d2)
print("Tens: ", d1)
print("Ones: ", d0)
print("Mirror Num: ", d0*100 + d1*10 + d2)
```

תרגיל מס' 32.

תפענח את הקוד הבא והצג דוגמה לפלט האפשרי של הקוד.

```
import random
N = random.randrange(100,1000)
print("Num: ", N)
d2 = int(N/100)
d1 = int((N-d2*100)/10)
d0 = N%10
print("Hundreds: ", d2)
print("Tens: ", d1)
print("Ones: ", d0)
print("Mirror Num: ", d1*100 + d0*10 + d2)
```



תרגיל מס' 33.

תפענח את הקוד הבא והצג דוגמה לפלט האפשרי של הקוד.

```
import random
N = random.randrange(100,1000)
print("Num: ", N)
d2 = int(N/100)
d1 = int((N-d2*100)/10)
d0 = N%10
print("Hundreds: ", d2)
print("Tens: ", d1)
print("Ones: ", d0)
print("Mirror Num: ", d0*100 + d2*10 + d1)
```

תרגיל מס' 34.

תפענח את הקוד הבא והצג דוגמה לפלט האפשרי של הקוד.

```
import random
N = random.randrange(100,1000)
print("Num: ", N)
d2 = int(N/100)
d1 = int((N-d2*100)/10)
d0 = N%10
print("Hundreds: ", d2)
print("Tens: ", d1)
print("Ones: ", d0)
print("Mirror Num: ", d1*100 + d2*10 + d0)
```

תרגיל מס' 35.

תפענח את הקוד הבא והצג דוגמה לפלט האפשרי של הקוד.

```
import random
N = random.randrange(100,1000)
print("Num: ", N)
d2 = int(N/100)
d1 = int((N-d2*100)/10)
d0 = N%10
print("Hundreds: ", d2)
print("Tens: ", d1)
print("Ones: ", d0)
print("Mirror Num: ", d2*100 + d0*10 + d1)
```

תרגיל מס' 36.

תפענח את הקוד הבא והצג דוגמה לפלט האפשרי של הקוד.

```
import random
N = random.randrange(0,86400)
print("Number of seconds: ", N)
h=int(N/3600)
print("Hours: ",h)
m = int((N-h*3600)/60)
print("Minutes: ", m)
sec=N-h*3600-m*60
print("Seconds: ",sec)
```

תרגילים בנושא טבלאות מעקב.

תרגיל מס' 1.

תפענח את הקוד הבא. מהו פלט של התוכנית, במקרה שקלט הוא 121, 7?

```
apple=int(input("Enter the number of apples: "))
children=int(input("Enter the number of children: "))
each=int(apple/children)
basket=apple%children
print(" Each child will receive " + str(each) + " apples.")
print(" In basket wil remain " + str(basket) + " apples.")
```

תרגיל מס' 2.

תפענח את הקוד הבא. מהו פלט של התוכנית, במקרה שקלט הוא 161, 23?

```
pupils = input("Num of pupils: ")
apples = input("Num of apples: ")
pupils = int(pupils)
apples = int(apples)
apples_on_pupil = apples // pupils
apples_in_basket = apples % pupils
print("Apples for each one: ", apples_on_pupil)
print("Apples in basket: ", apples_in_basket)
```

תרגיל מס' 3.

תפענח את הקוד הבא. מהו פלט של התוכנית, במקרה שקלט הוא 7 ?

```
let=int(input("Enter the number of the letter: "))
letter=chr(65+let-1)
print(letter)
```

תרגיל מס' 4.

תפענח את הקוד הבא. מהו פלט של התוכנית, במקרה שקלט הוא 15 ?

```
number = input("Enter the letter ")
number = int(number)
first_letter_code = ord('A')
your_letter_code = first_letter_code + number - 1
your_letter = chr(your_letter_code)
print("This is {}".format(your_letter))
```

תרגיל מס' 5.

תפענח את הקוד הבא. מהו פלט של התוכנית, במקרה שקלט הוא 4,5 ?

```
import math
a=int(input("Enter the side: "))
b=int(input("Enter another side: "))
c=math.sqrt(a*a+b*b)
print("Side c is: " + str(c))
square=a*b/2
p=a+b+c
print(" Square is " + str(square) + " cm².")
print(" Perimeter " + str(p) + " cm.")
```

תרגיל מס' 6.

תפענח את הקוד הבא. מהו פלט של התוכנית, במקרה שקלט הוא 5,3 ?

```
import math
AB = input("first side: ")
AC = input("second side ")
AB = float(AB)
AC = float(AC)
BC = math.sqrt(AB**2 + AC**2)
S = (AB * AC) / 2
P = AB + AC + BC
print("Third side is: %.2f " %BC)
print("Perimeter is: %.2f " %P)
print("Square is: %.2f " %S)
```

תרגיל מס' 7.

תפענח את הקוד הבא. מהו פלט האפשרי של התוכנית, במקרה שקלט הוא 50,30 ?

```
import random
a=int(input("Enter minimum number: "))
b=int(input("Enter maximum number: "))
c=random.randint(a,b)
print("Random number is " + str(c))
```

תרגיל מס' 8.

תפענח את הקוד הבא. מהו פלט האפשרי של התוכנית?

```
import random
c=round(random.random(),2)
print("Random number is :")
print(c)
```

תרגיל מס' 9.

תפענח את הקוד הבא. מהו פלט האפשרי של התוכנית?

```
from random import random
a = 0.1
b = 0.5
n = random() * (b - a) + a
print(round(n, 2))
print("%.2f" % n)
```

תרגיל מס' 10.

תפענח את הקוד הבא. מהו פלט האפשרי של התוכנית?

```
import random
sums=0
c=random.randint(100,999)
print(c)
sums=c%10+int(c/100)+int(c/10)%10
print(sums)
```

תרגיל מס' 11.

תפענח את הקוד הבא. מהו פלט האפשרי של התוכנית?

```
from random import randint
n = randint(100, 999)
print("Random number is:", n)
d1 = n % 10
d2 = n % 100 // 10
d3 = n // 100
print("Sum of cyphers ", d1 + d2 + d3)
```

תרגיל מס' 12.

תפענח את הקוד הבא. מהו פלט האפשרי של התוכנית?

```
from random import randint
n = randint(100, 999)
print("Random number is:", n)
d1 = int(n[0])
d2 = int(n[1])
d3 = int(n[2])
print("Sum of cyphers ", d1 + d2 + d3)
```

תרגיל מס' 13.

תפענח את הקוד הבא. מהו פלט האפשרי של התוכנית, במקרה שקלט הוא 2?

```
import math
rad=int(input("Please enter the radius: "))
p=2*math.pi*rad
print("Length is "+str(p))
s=math.pi*rad*rad
print("The square is " + str(s))
```

תרגיל מס' 14.

תפענח את הקוד הבא. מהו פלט האפשרי של התוכנית, במקרה שקלט הוא 3 ו- 2?

```
import math
rad=int(input("Please enter the radius: "))
height=int(input("Please enter the height: "))
s=2*math.pi*rad*height
print("The square is " + str(s))
```

תרגיל מס' 15.

תפענח את הקוד הבא. מהו פלט האפשרי של התוכנית, במקרה שקלט הוא 4 ו-5?

```
from math import pi
h = float(input('h = '))
r = float(input('r = '))
circles = 2 * (pi * r**2)
side = 2 * pi * r * h
area = circles + side
print('A =', round(area, 2))
```

תרגיל מס' 16.

תפענח את הקוד הבא. מהו פלט האפשרי של התוכנית, במקרים הבאים של קלט:

- א. 4, 5, -2, -1, 3?  
ב. -4, -3, -1, 4, -2?

```
import math
x1=int(input("Please enter the x of first point: "))
y1=int(input("Please enter the y of first point: "))
x2=int(input("Please enter the x of second point: "))
y2=int(input("Please enter the y of second point: "))
k=(y2-y1)/(x2-x1)
print(k)
angle=math.atan(k)*180/math.pi
print(angle)
b=y1-k*x1
print("y = " +str(k)+"x + " + str(b))
y=b
x=-b/k
print("line x at " + str(x))
print("line y at " + str(y))
```

תרגיל מס' 17.

תפענח את הקוד הבא. מהו פלט האפשרי של התוכנית, במקרים הבאים של קלט:

- א. -3, 2, -2, -1, -4?  
ב. 1, -3, -2, -1, -2?

```
print(" Give the point A(x1; y1)")
x1 = float(input(" \tx1 = "))
y1 = float(input(" \ty1 = "))
```

```

print("Give the point B(x2; y2)")
x2 = float(input(" \t x2 = "))
y2 = float(input(" \t y2 = "))
print(" Equation of the line is:")
k = (y1 - y2) / (x1 - x2)
b = y2 - k*x2
print(" y = %.2f*x + %.2f" % (k, b))

```

תרגיל מס' 18.

תפענח את הקוד הבא. מהו פלט האפשרי של התוכנית, במקרים הבאים של קלט:

א. 151 ו- 65?

ב. 74 ו- 91?

```

a=int(input("Please enter first number: "))
b=int(input("Please enter second number: "))
print(bin(a))
print(bin(b))
print("a and b = " + str( bin(a & b)))
print("a or b = " + str( bin(a | b)))
print("a xor b = " + str( bin(a ^ b)))

```

תרגיל מס' 19.

תפענח את הקוד הבא. מהו פלט האפשרי של התוכנית, במקרים הבאים של קלט:

א. 21 ו- 57?

ב. 34 ו- 91?

```

a=input("Please enter first number: ")
b=input("Please enter second number: ")
a=int(a, 2)
b=int(b, 2)
print(bin(a))
print(bin(b))
print("a and b = " + str( bin(a & b)))
print("a or b = " + str( bin(a | b)))
print("a xor b = " + str( bin(a ^ b)))

```

תרגיל מס' 20.

תפענח את הקוד הבא. מהו פלט האפשרי של התוכנית, במקרים הבאים של קלט:

א. 71 ו-65?

ב. 56 ו-37?

```
n1 = input("First: ")
n2 = input("Second: ")

n1 = int(n1, 2)
n2 = int(n2, 2)

bit_or = n1 | n2
bit_and = n1 & n2
bit_xor = n1 ^ n2

print(" OR: %s" % bin(bit_or))
print("AND: %s" % bin(bit_and))
print("XOR: %s" % bin(bit_xor))
```

תרגיל מס' 21.

תפענח את הקוד הבא. מהו פלט האפשרי של התוכנית, במקרה של קלט 342?

```
n = input("Enter 3 digit number: ")
n = int(n)
d1 = n % 10
n = n // 10
d2 = n % 10
n = n // 10
print("Sum is:", n + d1 + d2)
```

תרגיל מס' 22.

תפענח את הקוד הבא. מהו פלט האפשרי של התוכנית, במקרה של קלט 158?

```
n = input("Enter 3 digit number: ")
n = int(n)
d1 = n % 10
d2 = n % 100 // 10
d3 = n // 100
print("Sum is:", d1 + d2 + d3)
```



תרגיל מס' 23.

תפענח את הקוד הבא. מהו פלט האפשרי של התוכנית, במקרה של קלט 5?

```
from math import sqrt
n = input("Enter the number: ")
n = int(n)
print(n*4)
print(n**2)
print(n*sqrt(2))
```

תרגיל מס' 24.

תפענח את הקוד הבא. מהו פלט האפשרי של התוכנית, במקרה שקלט הוא 3 ו-4?

```
from math import sqrt
n = input("Enter first number: ")
n = int(n)
m = input("Enter second number: ")
m = int(m)
print(2*(n+m))
print(n*m)
print(sqrt(n**2+m**2))
```

תרגיל מס' 25.

תפענח את הקוד הבא. מהו פלט האפשרי של התוכנית, במקרה שקלט הוא 1000?

```
n = 1.06
initial=int(input("Enter the initial sum: "))
initial*=n
print("After 1st year: "+str(initial))
initial*=n
print("After 2nd year: "+str(initial))
initial*=n
print("After 3rd year: "+str(initial))
```

תרגיל מס' 26.

תפענח את הקוד הבא. מהו פלט האפשרי של התוכנית, במקרה שקלט הוא 30 ו-60?

```
a = float(input("Please Enter the First Angle of a Triangle: "))
b = float(input("Please Enter the Second Angle of a Triangle: "))
c = 180 - (a + b)
print("Third Angle of a Triangle = ", c)
```

תרגיל מס' 27.

תפענח את הקוד הבא. תן דוגמה לפלט האפשרי של התוכנית?

```
import random
a = random.randrange(1,10)
print("Side of cube is: ", a)
V = a**3
print("Volume: ", V)
S = 6 * a**2
print("Square of sides is: ", S)
```

תרגיל מס' 28.

תפענח את הקוד הבא. תן דוגמה לפלט האפשרי של התוכנית?

```
a = random.randrange(1,10)
b = random.randrange(1,10)
c = random.randrange(1,10)
print("Box ")
print("Side a: ", a)
print("Side b: ", b)
print("Height c: ", c)
V = a*b*c
print("Volume: ", V)
S = 2*(a*b + b*c + a*c)
print("Sum of squares: ", S)
```

תרגיל מס' 29.

תפענח את הקוד הבא. תן דוגמה לפלט האפשרי של התוכנית?

```
import random
from math import sqrt
a = random.randrange(1,10)
b = random.randrange(1,10)
print("a = ", a)
print("b = ", b)
print("Middle point ", (a+b)/2)
print("Geometric point ", sqrt(a*b))
a2 = a**2
b2 = b**2
print("Square a = ", a2)
print("Square b = ", b2)
print("Sum of Squares: ", a2+b2)
```

```
print("Difference of Squares: ", abs(a2-b2))
print("Multiplication of Squares: ", a2*b2)
print("Division of Squares: ", a2/b2)
```

תרגיל מס' 30.

תפענח את הקוד הבא. תן דוגמה לפלט האפשרי של התוכנית?

```
import random
a = random.uniform(-100,100)
b = random.uniform(-100,100)
print("a = ", a)
print("b = ", b)
abs_a = abs(a)
abs_b = abs(b)
print("Absolute value of a = ", abs_a)
print("Absolute value of b = ", abs_b)
print("Sum of Absolute values: ", abs_a + abs_b)
print("Difference of Absolute values: ", abs(abs_a - abs_b))
print("Multiplication of Absolute values: ", abs_a * abs_b)
print("Division of Absolute values: ", abs_a / abs_b)
```

תרגיל מס' 31.

תפענח את הקוד הבא. תן דוגמה לפלט האפשרי של התוכנית?

```
import random
import math
print("Pi:", math.pi)
R1 = random.randrange(2,10)
R2 = random.randrange(1,R1)
print("Radius 1: ", R1)
print("Radius 2: ", R2)
S1 = math.pi * R1**2
S2 = math.pi * R2**2
S3 = S1 - S2
print("Square of circle 1: ", S1)
print("Square of circle 2: ", S2)
print("Square of disk: ", S3)
```

תרגיל מס' 32.

תפענח את הקוד הבא. תן דוגמה לפלט האפשרי של התוכנית?

```
import random
x1 = random.randrange(-10,10)
x2 = random.randrange(-10,10)
print("x1: ", x1)
print("x2: ", x2)
print("Distance: ", abs(x2-x1))
print("Half of the line is:", (x1+x2)/2)
```

תרגיל מס' 33.

תפענח את הקוד הבא. תן דוגמה לפלט האפשרי של התוכנית?

```
import random
A,C,B = sorted(random.sample(range(-10, 10), 3))
print("A: ", A)
print("B: ", B)
print("C: ", C)
AC = abs(A-C)
BC = abs(B-C)
AC_BC = AC + BC
AB = abs(A-B)
print("AC: ", AC)
print("BC: ", BC)
print("AC + BC: ", AC_BC)
print("AB: ", AB)
AC_BC = AC * BC
print("AC * BC: ", AC_BC)
```

תרגיל מס' 34.

תפענח את הקוד הבא. תן דוגמה לפלט האפשרי של התוכנית?

```
import random
x1,x2 = random.sample(range(-10, 10), 2)
y1,y2 = random.sample(range(-10, 10), 2)
print("(x1, y1): ({0},{1})".format(x1, y1))
print("(x2, y2): ({0},{1})".format(x2, y2))
side1 = abs(x1 - x2)
side2 = abs(y1 - y2)
P = 2 * (side1 + side2)
S = side1 * side2
```

```
print("Side 1: ", side1)
print("Side 2: ", side2)
print("Perimeter: ", P)
print("Square: ", S)
```

תרגיל מס' 35.

תפענח את הקוד הבא. תן דוגמה לפלט האפשרי של התוכנית?

```
import random
import math
x1,x2 = random.sample(range(-10, 10), 2)
y1,y2 = random.sample(range(-10, 10), 2)
print("Point 1 (x1, y1): ({0},{1})".format(x1, y1))
print("Point 2 (x2, y2): ({0},{1})".format(x2, y2))
side1 = abs(x1 - x2)
side2 = abs(y1 - y2)
print("Side 1: ", side1)
print("Side 2: ", side2)
d = math.sqrt((x1 - x2)**2 + (y1 - y2)**2)
print("Distance: ", d)
```

תרגיל מס' 36.

תפענח את הקוד הבא. תן דוגמה לפלט האפשרי של התוכנית?

```
import math
x1,x2,x3,y1,y2,y3 = [0,3,0,0,0,4]
print(" A (x1, y1): ({0},{1})".format(x1, y1))
print(" B (x2, y2): ({0},{1})".format(x2, y2))
print(" C (x3, y3): ({0},{1})".format(x3, y3))
d_AB = math.sqrt((x1 - x2)**2 + (y1 - y2)**2)
print("AB: ", d_AB)
d_AC = math.sqrt((x1 - x3)**2 + (y1 - y3)**2)
print(" AC: ", d_AC)
d_BC = math.sqrt((x2 - x3)**2 + (y2 - y3)**2)
print(" BC: ", d_BC)
p = (d_AB + d_AC + d_BC)
print("P: ", p)
p/=2;
S = math.sqrt(p*(p-d_AB)*(p-d_AC)*(p-d_BC))
print("Square: ", S)
```

תרגיל מס' 37.

תפענח את הקוד הבא. תן דוגמה לפלט האפשרי של התוכנית?

```
import random
A,B = sorted(random.sample(range(-10, 10), 2))
print("A = {0}, B = {1}".format(A,B))
A,B = B,A
print("A = {0}, B = {1}".format(A,B))
A = B + A
B = A - B
A = A - B
print("A = {0}, B = {1}".format(A,B))
X = B
B = A
A = X
print("A = {0}, B = {1}".format(A,B))
```

תרגיל מס' 38.

תפענח את הקוד הבא. תן דוגמה לפלט האפשרי של התוכנית?

```
import random
A,B,C = sorted(random.sample(range(-10, 10), 3))
print("A = {0}, B = {1}, C = {2}".format(A,B,C))
A,B = B,A
A,C = C,A
print("A = {0}, B = {1}, C = {2}".format(A,B,C))
```

תרגיל מס' 39.

תפענח את הקוד הבא. תן דוגמה לפלט האפשרי של התוכנית?

```
import random
A,B,C = sorted(random.sample(range(-10, 10), 3))
print("A = {0}, B = {1}, C = {2}".format(A,B,C))
A,C = C,A
A,B = B,A
print("A = {0}, B = {1}, C = {2}".format(A,B,C))
```

תרגיל מס' 40.

תפענח את הקוד הבא. תן דוגמה לפלט האפשרי של התוכנית?

```
import random
x = random.randrange(-10, 10)
print("x = ",x)
y = 3*x**6 - 6*x**2 - 7
print("3*{0}^6 - 6*{0}^2 - 7 = {1}".format(x,y))
```

תרגיל מס' 41.

תפענח את הקוד הבא. תן דוגמה לפלט האפשרי של התוכנית?

```
import random
x = random.randrange(-10, 11)
print("x = ",x)
t = (x-3)**3
y = 4*t**2 - 7*t + 2
print("4*({0}-3)^6 - 7*({0}-3)^3 + 2 = {1}".format(x,y))
```

תרגיל מס' 42.

תפענח את הקוד הבא. תן דוגמה לפלט האפשרי של התוכנית?

```
import random
import math
degree = random.randrange(0, 360)
degree = 180
radian = degree / 180 * math.pi
print("Degree = ",degree)
print("Radians = ",radian)
```

תרגיל מס' 43.

תפענח את הקוד הבא. תן דוגמה לפלט האפשרי של התוכנית?

```
import random
import math
radian = random.uniform(0, 2*math.pi)
#radian = math.pi/2
degree = radian * 180 / math.pi
print("Radians = ",radian)
print("Degree = ",degree)
```

תרגיל מס' 44.

תפענח את הקוד הבא. תן דוגמה לפלט האפשרי של התוכנית?

```
import random
X,Y = sorted(random.sample(range(1, 10), 2))
A = random.randrange(3000,10000)
B = random.randrange(1000,3000)
price_X = A/X
price_Y = B/Y
print(X,"kg of chocolates = ",A," dollar ")
print(Y,"kg of toffees = ",B," dollar ")
print("1 kg of chocolates = {0:.2f} dollar".format(price_X))
print("1 kg of toffees = {0:.2f} dollar ".format(price_Y))
print("Chocolate candy is more expensive \than toffee, times:",round(price_X/price_Y))
```

תרגיל מס' 45.

תפענח את הקוד הבא. תן דוגמה לפלט האפשרי של התוכנית?

```
import random
U,V = sorted(random.sample(range(10, 21), 2))
T1,T2 = sorted(random.sample(range(1, 10), 2))
Distance_Lake = V * T1
Distance_River = (V - U) * T2
print("Speed in the lake:",V)
print("Speed of the river:",U)
print("Speed in the river (up the stream):",V-U)
print("Time in the lake:",T1)
print("Time in the river:",T2)
print("Path in the lake:",Distance_Lake)
print("Path in the river:",Distance_River)
```

תרגיל מס' 46.

תפענח את הקוד הבא. תן דוגמה לפלט האפשרי של התוכנית?

```
import random
V1,V2 = sorted(random.sample(range(50, 70), 2))
S = random.randrange(0,20)
T = random.randrange(1,6)
total_speed = V1 + V2
total_path = total_speed * T
D = S + total_path
```



```
print("1st vehicle speed:",V1)
print("2nd vehicle speed:",V2)
print("Time:",T)
print("Total speed:",total_speed)
print("Initial Distance between vehicles:",S)
print("Final Distance between vehicles:",D)
```

תרגיל מס' 47.

תפענח את הקוד הבא. תן דוגמה לפלט האפשרי של התוכנית?

```
import random
V1,V2 = sorted(random.sample(range(50, 70), 2))
S = random.randrange(400,600)
T = random.randrange(1,6)
total_speed = V1 + V2
total_path = total_speed * T
D = abs(S - total_path)
print("1st vehicle speed:",V1)
print("2nd vehicle speed:",V2)
print("Time:",T)
print("Total speed:",total_speed)
print("Initial Distance between vehicles:",S)
print("Final Distance between vehicles:",D)
```

תרגיל מס' 48.

תפענח את הקוד הבא. תן דוגמה לפלט האפשרי של התוכנית?

```
import random
A = random.randrange(-10,10)
B = random.randrange(-10,10)
x = -B/A
y = A * x + B
print("Ax + B = 0")
print("A = ",A)
print("B = ",B)
print("x = ",-B/A)
print("({0}) * ({1}) + ({2}) = {3}".format(A,x,B,y))
```

תפענח את הקוד הבא. תן דוגמה לפלט האפשרי של התוכנית?

```
import random
import math
A1 = random.randrange(-10,10)
B1 = random.randrange(-10,10)
A2 = random.randrange(-10,10)
B2 = random.randrange(-10,10)
C1 = random.randrange(-10,10)
C2 = random.randrange(-10,10)
D = A1*B2 - A2*B1
print("A1 = ",A1)
print("B1 = ",B1)
print("C1 = ",C1)
print()
print("A2 = ",A2)
print("B2 = ",B2)
print("C2 = ",C2)
print("D = ",D)
print()
print("{0}x + {1}y = {2}".format(A1,B1,C1))
print("{0}x + {1}y = {2}".format(A2,B2,C2))

x = (C1*B2 - C2*B1)/D
y = (A1*C2 - A2*C1)/D
print("x = ",x)
print("y = ",y)
print("Check Equation 1:",A1*x+B1*y)
print("Check Equation 2:",A2*x+B2*y)
```

## תרגילים בנושא כתיבת קוד.

תרגיל מס' 1.

קולטים מספר תפוחים שיש בתוך הסל וכמות ילדים שרוצים לקחת תפוחים ותרשום כמה תפוחים כל ילד יכול לקחת וכמה נשאר בתוך הסל.

תרגיל מס' 2.

קולטים מספר (מיקום) של האות באלף-בית האנגלי מהמקלדת. הצגת התו המבוקש.

תרגיל מס' 3.

מצא שטח של משולש ישר זווית לפי שני ניצבים.

תרגיל מס' 4.

מצא הקף ושטח של משולש ישר זווית לפי שני ניצבים (דרך משפט פיתגורס).

תרגיל מס' 5.

מהמקלדת מקבלים גבולות לטווח מספרי. הפעל גנרטור של מספר אקראי שלם בתוך התחום והצג אותו.

תרגיל מס' 6.

קבל מספר עשרוני אקראי בטווח של 0.10 עד 0.50. הצג אותו על המסך עם שני מקומות עשרוניים.

תרגיל מס' 7.

כתוב תוכנית שמייצרת מספר אקראי בן שלוש ספרות ומחשבת את סכום הספרות שלו.

תרגיל מס' 8.

בעזרת הרדיוס שהוזן מהמקלדת, חשב את אורך המעגל המתאים ואת שטח המעגל.

תרגיל מס' 9.

כתוב תוכנית המחשבת את שטח הפנים של הגליל בהתאם לרדיוס וגובה הגליל שנקלט מהמקלדת. שטח הפנים ניתן לחשב לפי נוסחה:  $S = 2\pi rh$ .

תרגיל מס' 10.

כתוב תוכנית המציגה את המשוואה של הקו  $y = kx + b$  העוברת בנקודות נתונות. קואורדינטות הנקודות מוזנות מהמקלדת. הדפס גם זווית של הקו ובאיזה נקודות קו חוצה את הצירים.

שאלה מס' 11.

מהמקלדת מוזנים שני מספרים בינאריים. נדרש להציג בצורה בינארית את התוצאות של פעולות AND, OR, EXCLUSIVE OR על המספרים האלה.

שאלה מס' 12.

מצא את סכום הספרות של מספר תלת-ספרתי שנקלט מהמשתמש.

שאלה מס' 13.

כתוב תוכנית שמקבלת מהמשתמש צלע של ריבוע ומדפיסה: היקף הריבוע, שטח הריבוע ואורך של האלכסון העובר בו.

שאלה מס' 14.

כתוב תוכנית שמקבלת מהמשתמש צלעות של מלבן ומדפיסה: היקף המלבן, שטח המלבן ואורך של האלכסון העובר בו.

שאלה מס' 15.

משתמש מכניס סכום מסוים בבנק ומקבל תוספת 6% כל שנה. תקלוט מהמשתמש סכום ראשון ותדפיס מהו סכום שיהיה בחשבון לאחר שנה, שנתיים, שלוש שנים.

שאלה מס' 16.

כתוב תוכנית שמייצרת מספר אקראי בגבולות בין 6 לבין 12.

שאלה מס' 17.

כתוב תוכנית שמייצרת מספר אקראי בגבולות בין 5 לבין 100 בקפיצות של 5.

שאלה מס' 18.

כתוב תוכנית הקולטת מהמשתמש 2 מספרים ומחזירה תוצאה (TRUE או FALSE) אם הם שווים או לא.

(ללא משפט IF).

שאלה מס' 19.

כתוב תוכנית הקולטת מהמשתמש 3 מספרים ובודקת האם כולם שווים או שיש שתיים ששווים. (אין להשתמש במשפט IF).

שאלה מס' 20.

כתוב תוכנית הקולטת ציון של התלמיד וכותבת על המסך האם הוא עבר את המבחן (גדול מ-55) לא עבר את המבחן (קטן מ-55) או נמצא על הגבול (שווה ל-55), ללא שימוש במשפט IF.

שאלה מס' 21.

כתוב תוכנית שמחשבת משפט מתמטי (תוצאה צריכה להיות שלמה).

$$\frac{512 - 282}{47 \cdot 48 + 5}$$

שאלה מס' 22.

כתוב תוכנית שמקבלת ערך X וערך Y ומחשבת  $\frac{|x-y|}{x+y}$ .

שאלה מס' 23.

כתוב תוכנית שמקבלת זווית בין 180 לבין 180- מעלות ומחזירה תוצאה במעלות בין 0 ל- 360 מעלות וגם ברדיאנים בין 0 לבין 2π.

שאלה מס' 24.

כתוב תוכנית שמקבלת מספר שניות בצורה אקראית ( בין 10 000 לבין 1 000 000 ) ומחזירה את כמות ימים, שעות, דקות, שניות.

שאלה מס' 25 .

כתוב תוכנית הקולטת מהמשתמש מספר ובודקת האם הוא זוגי או אי-זוגי בעזרת אופרטורים של סיביות.

שאלה מס' 26.

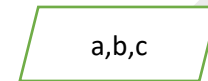
כתוב תוכנית הקולטת מהמשתמש מספר והופכת אותו מזוגי לאי-זוגי ואי-זוגי לזוגי בעזרת אופרטורים של סיביות.

שאלה מס' 27.

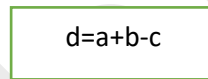
כתוב תוכנית שמפרקת מספר בן 4 ספרות ל-4 מספרים חד ספרתיים בעזרת אופרטורים של סיביות.

# פרק ג. ביצוע מותנה.

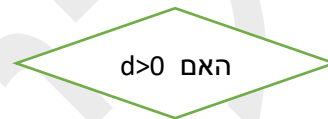
מבוא לאלגוריתמים.  
הגדרת של אירועים:  
התחלה וסיום:



קלט/פלט:



פעולה:

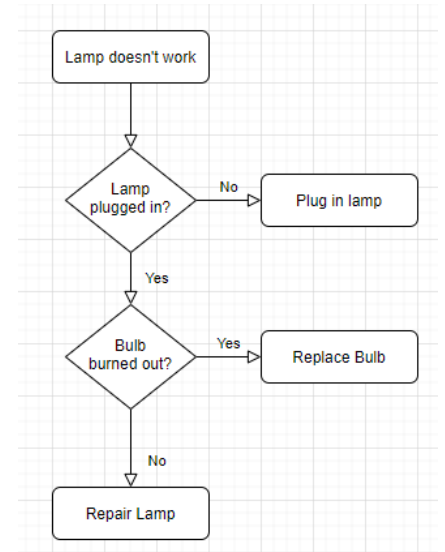
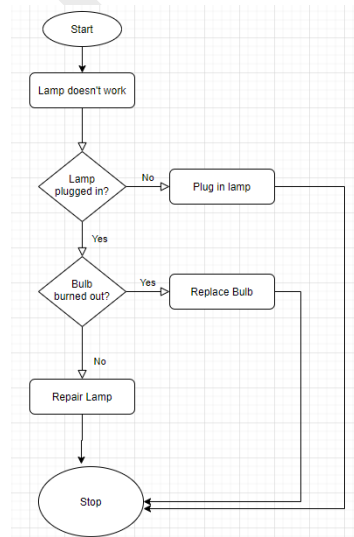


תנאי:

מספר כללים:

תוכנית מתחילה תמיד ב- Start ומסתיימת ב- Stop

אסור שאחד מהנתיבים ישאר באיור ללא סיום או ניתוב הלאה בקוד.  
דוגמה ללוגיקה נכונה וביצוע לא נכון: תיקון לתרשימי זרימה:

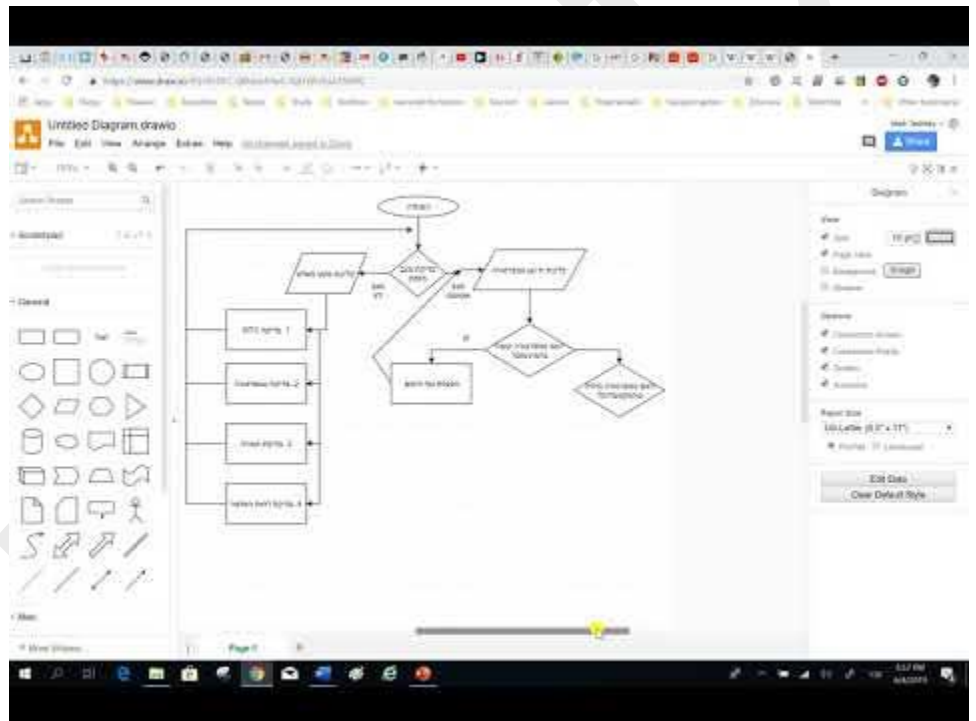


## תוכנת סרטוט תרשמי זרימה.

ישנה תוכנה חינמית באינטרנט בכתובת <https://app.diagrams.net/> כדי לבנות תרשימי זרימה (אלגוריתמים).

ניתן לראות סרטון הדרכה:

[https://www.youtube.com/watch?v=IYBXJPNVWg4&list=PLfOLrvQnsct1Th6fPx2s\\_sbIuvTxhecI9&index=12&t=300s](https://www.youtube.com/watch?v=IYBXJPNVWg4&list=PLfOLrvQnsct1Th6fPx2s_sbIuvTxhecI9&index=12&t=300s)



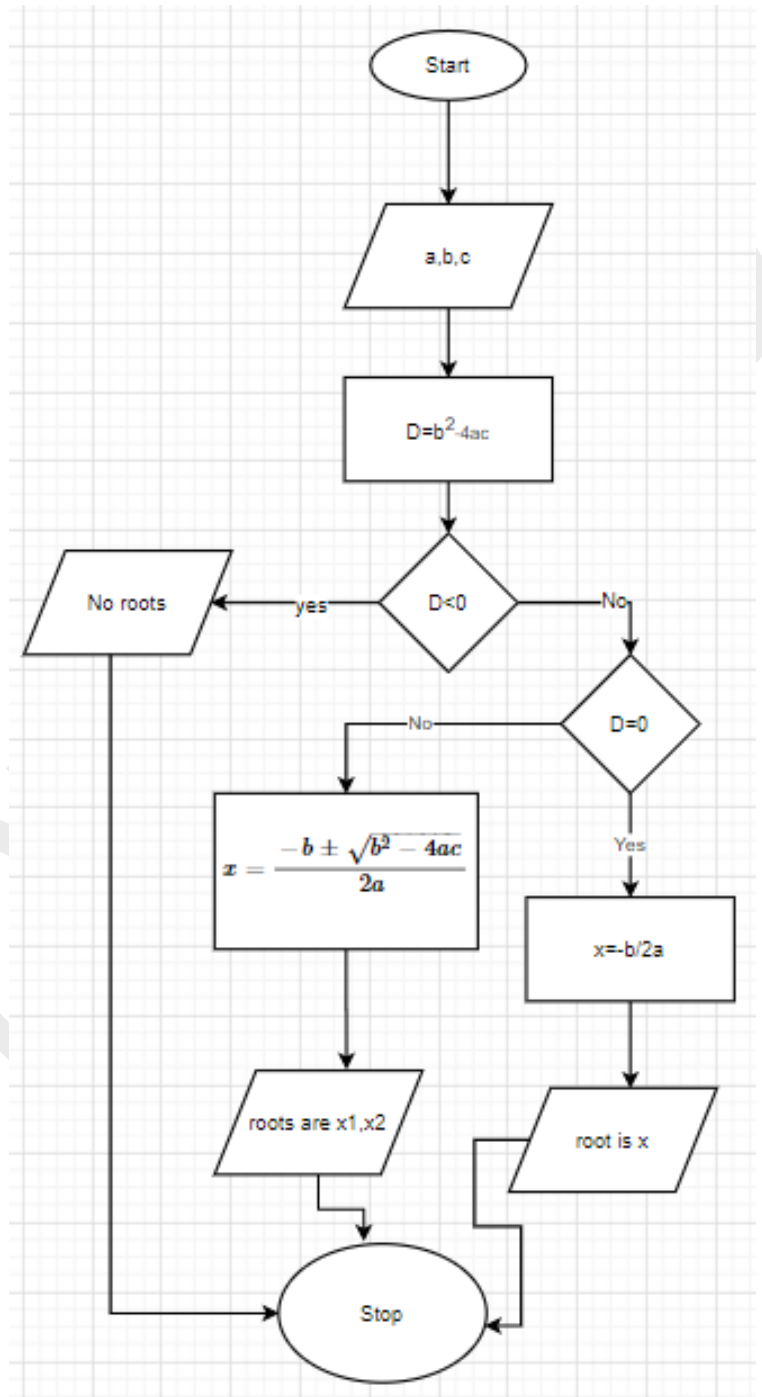
דוגמאות של תרשימי זרימה:  
אלגוריתם של פתרון של משוואה ריבועית:

קלט: a, b, c

חישוב:  $D = b^2 - 4ac$

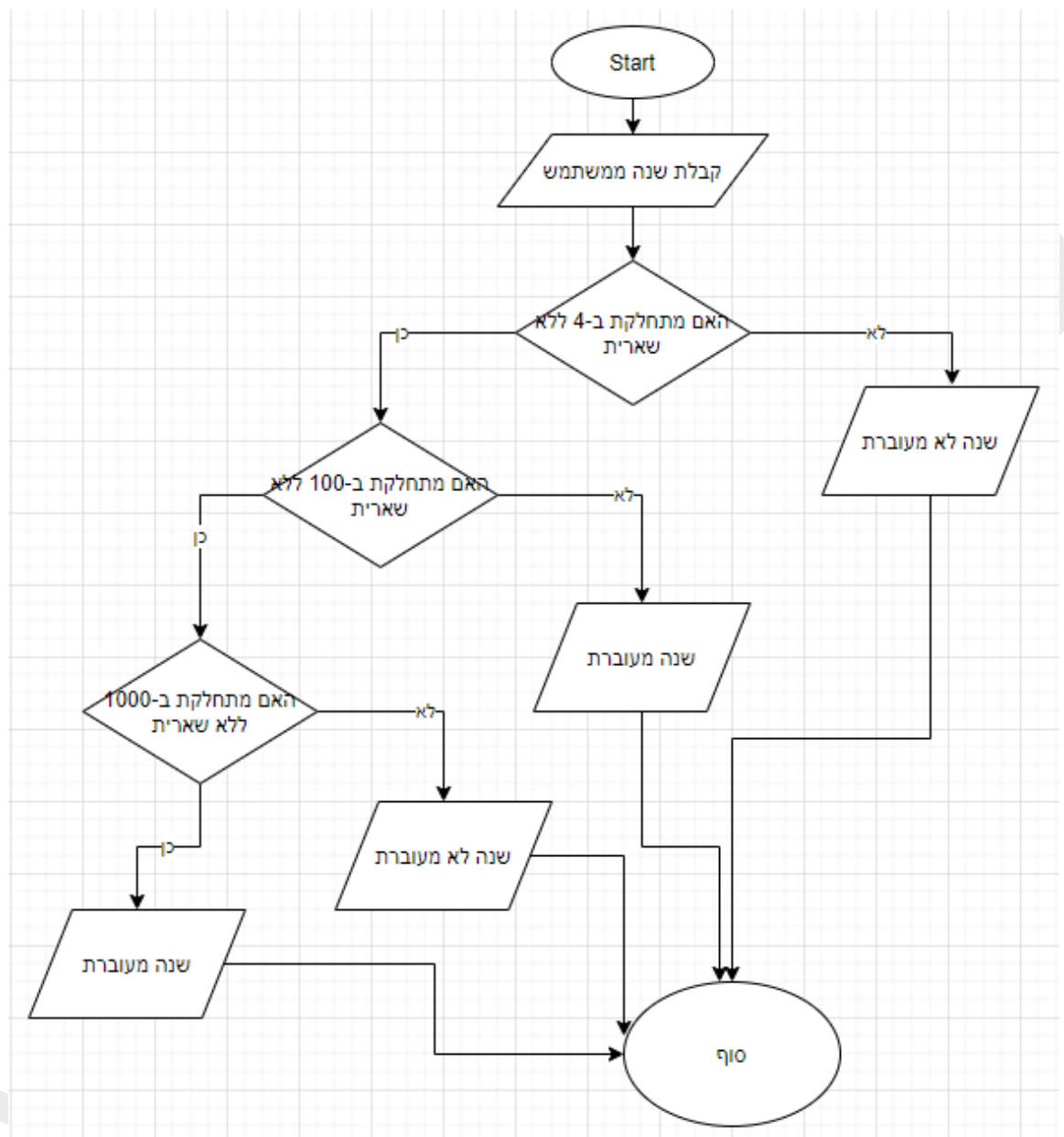
תנאים:  $D > 0, D = 0, D < 0$

פלט: x או  $x_1, x_2$



תרשימי זרימה לחיפוש שנה מעוברת.





## אופרטורים של התניה.

### אופרטור if.

התקדמות התוכנית יכולה להיות לינארית, כלומר ברצף התרחשויות שמתבצעות באופן כרונולוגי ונסיבתי, החל מהראשון ומסתיים באחרון. לא מדלגים אף שורה של קוד התוכנית. עם זאת, לעתים קרובות יותר בתוכניות, הדבר לא קורה באופן זה. בעת ביצוע הקוד, תלוי בתנאים אלה או אחרים, חלק מהקטעים עשויים להישמט, בעוד שאחרים יתממשו.

בואו נצייר אנלוגיה למציאות. האדם חי על פי לוח זמנים. ניתן לומר שלוח הזמנים הוא האלגוריתם של האדם, כלומר קוד התוכנית הכתוב לו לביצוע. בלוח הזמנים בשעה 18.00 הוא מתכנן ביקור בבריכה. עם זאת, בדקה 90 הוא מקבל מידע שבריכה סגורה, עקב פיצוץ בצנרת. יהיה זה נכון לבטל את הביקור בבריכה, כלומר לשנות את מסלול התוכנית בלוחות הזמנים לפי נסיבות קיימות שנובעות מהסביבה. אחד התנאים לביקור בבריכה צריך להיות תפקוד הבריכה, אחרת יש לבצע פעולות אחרות. נסיבתיות כזו של פעולות ניתנת ליישום בתוכנת מחשב. לדוגמה, חלק מהקוד יבוצע רק עבור ערך מסוים של משתנה ספציפי.

כתיבת תנאים תהיה בצורה הבאה:

בדוגמה של interpreter:

```
>>> b=0
>>> a=50
>>> n=98
>>> if n<100:
    b=n+a
>>> a
50
>>> b
148
```

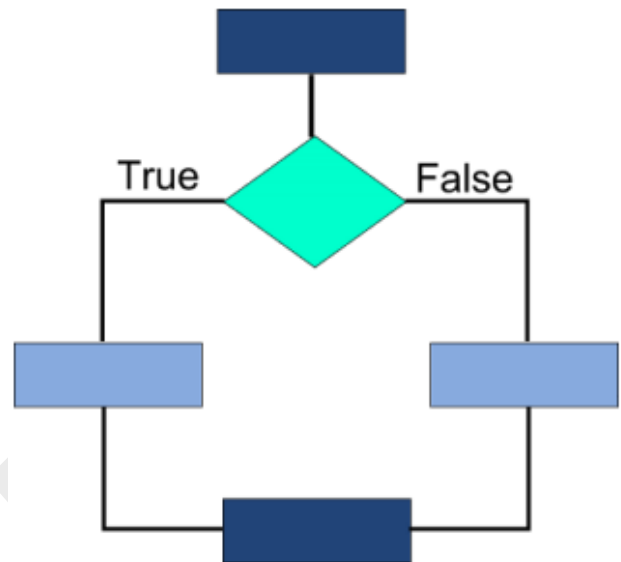
בדוגמה של סקריפט:

```
tv = 2500
dvd = 400
if tv+ dvd > 2800 :
    print("there are not enough money")
```

```
C:\Users\marka\AppData\Local\Programs\Python\Python38-32\python.exe C:/MyPython/third.py
there are not enough money
```

## אופרטור if else

במידה ויש פעולות גם במקרה של "כן" וגם במקרה של "לא" ניתן לציין אלגוריתם הבא:



במקרה כזה ניתן להשתמש באופרטור if else:

דוגמה פשוטה:

```
tv = int(input("tv"))
dvd = int(input("dvd"))
if tv+ dvd > 2800 :
    print("there are not enough money")
else:
    print("There are enough money")
```

```
C:\Users\marka\AppData\Local\Programs\Python\Python38-32\python.exe C:/MyPython/third.py
tv1200
dvd200
There are enough money
```

שימו לב שפונקציה input() תמיד מחזירה מחרוזת, אם לא נעשה המרה מתאימה, נקבל שגיאה.

```
tv = input("tv")
dvd = input("dvd")
if tv+ dvd > 2800 :
    print("there are not enough money")
else:
    print("There are enough money")
```

```
C:\Users\marka\AppData\Local\Programs\Python\Python38-32\python.exe C:/MyPython/third.py
tv1200
dvd200
Traceback (most recent call last):
  File "C:/MyPython/third.py", line 3, in <module>
    if tv+ dvd > 2800 :
TypeError: '>' not supported between instances of 'str' and 'int'
```

ניתן להתייחס למשתנים בצורת True/False, זאת אומרת ללא תנאים, כמו בדוגמה הבאה a מקבל פסוק אמת מסוים:

```
a=5>0
if a:
    print(a)
b=7
if b:
    print(b)
```

```
C:\Users\marka\AppData\Local\Programs\Python\Python38-32\python.exe C:/MyPython/third.py
True
7
```

שימו לב על משהו מעניין. במידה ותעשו <enter> ותכתבו תרגיל בצורה הבאה – תקבלו שגיאת קומפילציה:

```
a=5>0
if a:
    print(a)
b=7
if b:
    print(b)
```

```
C:\Users\marka\AppData\Local\Programs\Python\Python38-32\python.exe C:/MyPython/third.py
File "C:/MyPython/third.py", line 6
    print(b)
    ^
IndentationError: expected an indented block
```

צריכים להזיז את שורות 4 ו-5 שמאלה כדי לקמפל ולהריץ הוראות ללא שגיאה. בפיתוח קיימת צורה מיוחדת שלא הייתה בשפת C – במקום קישור תנאים בעזרת משפט OR יש אפשרות להשתמש בצורה מקוצרת:

```
a=5
b=10
if a>0 and a<b:
    print(b-a)
else: print(a-b)
```

5

צורה מקוצרת:

```
a=5
b=10
if 0<a<b:
    print(b-a)
else: print(a-b)
```

5

במידה ויש מספר תנאים שאחד יוצא מהשני, מסדרים אותם בעזרת "רווחים לבנים" ולא כמו בשפת C בעזרת {} לייצוג היררכי.

### **דוגמה: קואורדינטות x,y**

כתוב תוכנית הקולטת מהשתמש קואורדינטות x,y של נקודה ספציפית ומחזירה באיזה רבע של מישור נקודה נמצאת:

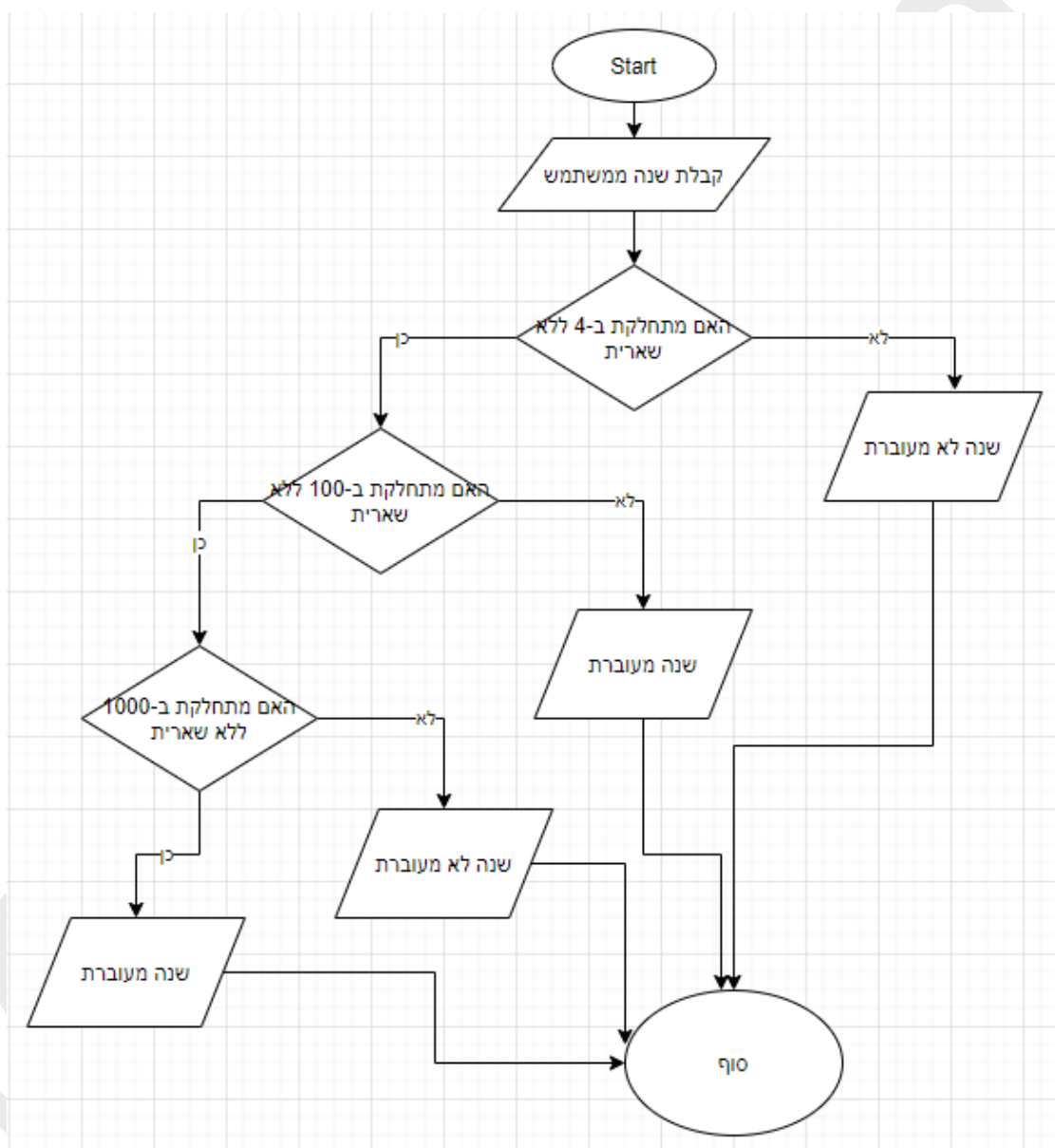
```
x = int(input("x="))
y = int(input("y="))
if x > 0:
    if y > 0: # x>0, y>0
        print("First quater")
    else: # x>0, y<0
        print("Fourth quater")
else:
    if y > 0: # x<0, y>0
        print("Second quater")
    else: # x<0, y<0
        print("Third quater")
```

```
x=2
y=-3
Fourth quater
```

## דוגמה: שנה מעוברת.

כתוב תוכנית שבודקת האם שנה היא שנה מעוברת או לא.

נתון אלגוריתם לבדיקה:



```
# Python program to check if year is a leap year or not
# To get year (integer input) from the user
year = int(input("Enter a year: "))
if (year % 4) == 0:
    if (year % 100) == 0:
        if (year % 400) == 0:
            print("{0} is a leap year".format(year))
        else:
            print("{0} is not a leap year".format(year))
    else:
        print("{0} is not a leap year".format(year))
else:
```

```
print("{0} is a leap year".format(year))
else:
print("{0} is not a leap year".format(year))
```

```
Enter a year: 1993
1993 is not a leap year
```

```
Enter a year: 1900
1900 is not a leap year
```

```
Enter a year: 2000
2000 is a leap year
```

```
Enter a year: 2020
2020 is a leap year
```

משפט "elif".  
להבדיל משפת C וכל היתר השפות, בשפת פייתון יש משפט מיוחד שנקרא "elif".  
ניקח דוגמה של שפת C ונעביר אותה לשפת פייתון:

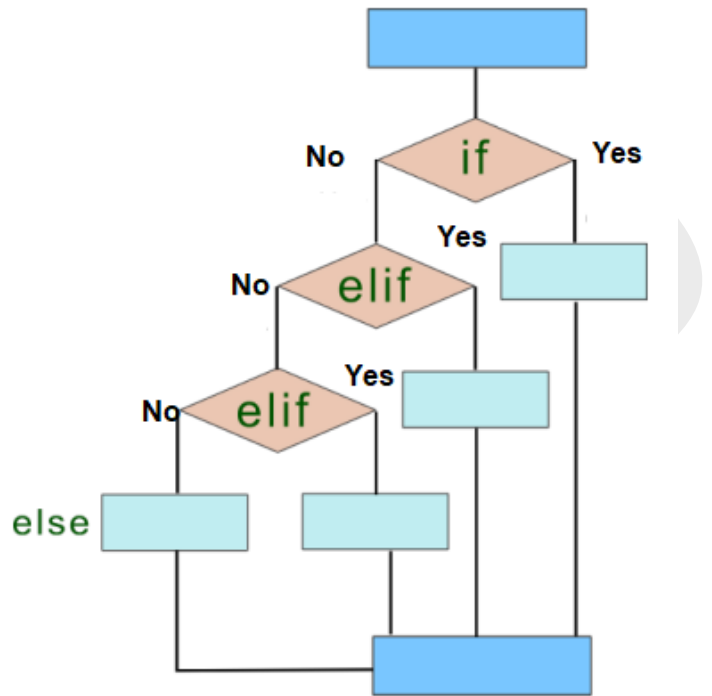
```
if(condition)
{
}

else if(condition)
{
}

else if(condition)
{
}

else
{
}
```

במקום else if בשפת פייתון (עצלנות של אנשי תוכנה 😊) כותבים את המילה elif. צורת העבודה של פעולה הזאת אפשר לתאר בסכמה הבאה:



נניח שאנו רוצים לכתוב תוכנית שמקבלת גיל של ילד ובהתאם לזה מחזירה סוג של משחק המומלץ לגיל שלו:

```
old = int(input('Your age '))
print('Toys for You:', end=' ')
if 3 <= old < 6:
    print("Cat")
elif 6 <= old < 12:
    print("Puzzle")
elif 12 <= old < 16:
    print("Lego")
elif 16 <= old <= 50:
    print("Arduino")
else: print("Any one from all")
```

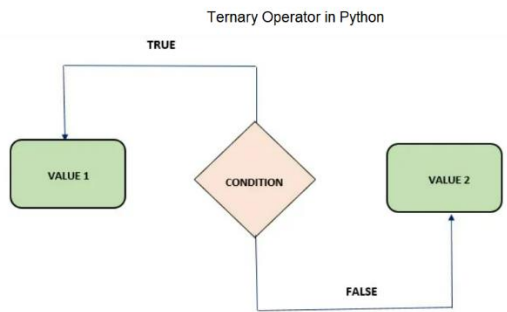
```
Your age 14
Toys for You: "Lego"
Your age 50
Toys for You: Any one from all
```



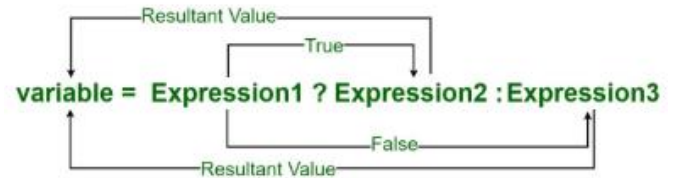
## אופרטור שורה if-else

אופרטורים האלה ידועים גם כביטויים מותנים. הם אופרטורים שמעריכים משהו על סמך תנאי שהוא נכון או לא נכון. זה התווסף לפייתון בגרסה 2.5.

זה פשוט מאפשר בדיקת תנאי בשורה אחת המחליף את ה-multiline if-else מה שהופך את הקוד לקומפקטי. אופרטור זה דומה לאופרטור "?" של שפת C:

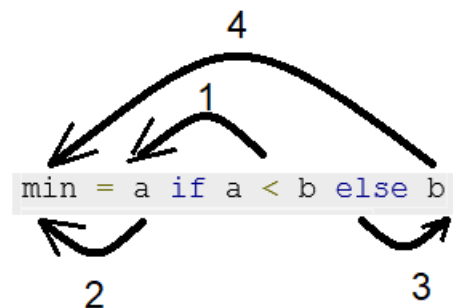


## Conditional or Ternary Operator (?:) in C/C++



לדוגמה:

```
a=int(input("a= "))
b=int(input("b= "))
min = a if a < b else b
print("min = ",min)
```



ניתן לתרגם משפט זה בצורה הבאה: אם  $a < b$ , אז  $a$  נכנס לתוך  $min$ , אחרת  $b$  נכנס לתוך  $min$ . אפשר לכתוב קטע קוד זה בצורה קצת אחרת:

```
a=int(input("a= "))
b=int(input("b= "))
print(a,"is greater") if (a>b) else print(b,"is Greater")
```

ניתן לשרשר תנאים ולרשום אותם בשורה אחד.

```
a=int(input("a= "))
b=int(input("b= "))
print ("equal" if a == b else "a is greater" if a > b else "b is greater")
```

```
a= 2
b= 2
equal
```

```
a= 5
b= 3
a is greater
```

```
a= 3
b= 4
b is greater
```

נראה השוואה בין אופרטור מותנה רגיל מול אופרטור שורה:

```
num=int(input("Enter number to check :"))
if(num%2==0):
    msg=str(num)+" is an even number"
else:
    msg=str(num)+" is an odd number"
print(msg)
```

```
num=int(input("Enter number to check :"))
msg=str(num)+" even number" if num%2==0 else str(num)+" odd number"
print(msg)
```

אותה פעולה, אך במקום 6 שורות תוכנה לקחה 3. הבדל משמעותי, יותר מזה תנאי עצמו התקצר מ-4 שורות לשורה 1.

לפני גרסה 2.5 הייה קיים אופרטור אחר שאפשר אותם פעולות כמו if else:

```
a=int(input("a= "))
b=int(input("b= "))
min = a < b and a or b
print(min)
```

הוא מבצע השוואה: האם  $a < b$ , אם כן מכניס a לתוך min, ואם לא מכניס את b לתוך min.

עם הופעת אופרטור השוואה צורה הישנה עדיין קיימת ופועלת.

אף אחד לא מחייב אותכם לכתוב כך, אבל אתם חייבים להכיר את צורת הכתיבה הזאת, כדי להבין אותה, אם תוכנית כתובה בפורמט הישן.

## משפט switch-case:

עד הגרסה Python 3.10 בשפת פיתון לא הייה משפט switch-case, לעומת זאת יש אפשרות לבנות את המבנה שדומה ל- switch-case. נראה דוגמה הבאה:

```
i = str(input("Enter the number between 1-7: "))
week_days = {
    '1': 'Sunday',
    '2': 'Monday',
    '3': 'Tuesday',
    '4': 'Wednesday',
    '5': 'Thursday',
    '6': 'Saturday',
    '7': 'Sunday'
}
print(week_days[i])
```

```
Enter the number between 1-7: 3
Tuesday
```

לפי מה שאנו רואים כאן, חסר מנגנון של default. ניתן לממש אותו בעזרת משפטי try, except:

```
K = str(input("Enter the number between 1-5: "))
mark = {
    '1': 'Troll',
    '2': 'Bad',
    '3': 'Not Bad',
    '4': 'Good',
    '5': 'Excellent'
}
try:
    print(mark[K])
except KeyError as e:
    print('error')
```

```
Enter the number between 1-5: 4
Good
Enter the number between 1-5: 6
error
```

ניתן להחליף קוד של switch and case ברצף הבא:

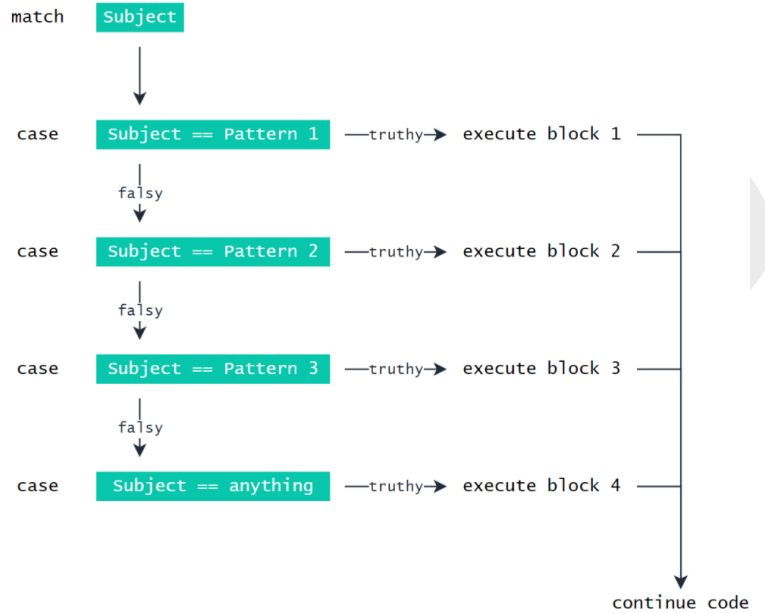
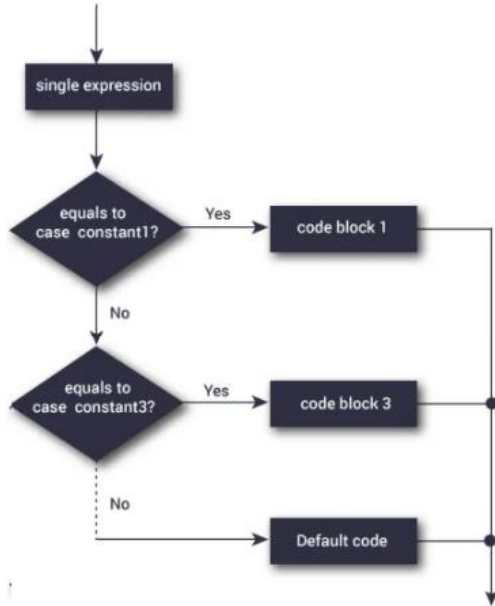
```
http_code = "418"
if http_code == "418":
    print("OK")
    do_something_good()
elif http_code == "404":
    print("Not Found")
    do_something_bad()
elif http_code == "418":
    print("I'm a teapot")
    make_coffee()
else:
    print("Code not found")
```

החל מגרסה Python 3.10 ישנה אפשרות להשתמש במשפט match case שדומה למשפט switch case של שפת C:

```
http_code = input("Enter the code ")
match http_code:
    case "200":
        print("OK")
    case "404":
        print("Not Found")
    case "418":
        print("I'm a teapot")
    case _:
        print("Code not found")
```

```
Enter the code 418
I'm a teapot
```

נראה את המבנה של הפקודה בהשוואה לשפת C:



שימו לב שבשפת פיתון משפט default לא קיים, במקום זה משתמשים במקרה:

case \_:

ניתן לשלב מספר תנאים ביחד בעזרת אופרטור "|".

```
status_code = int(input("Enter the code: "))
match status_code:
    case 400|401|403 :
        print("bad request")
    case 200:
        print("good")
    case _:
        print("Something else bad happened")
```

```
Enter the code: 400
bad request
Enter the code 403
bad request
Enter the code 12
Something else bad happened
```

דוגמה נוספת למימוש משפט:

```
sample = True
match sample:
    case (True | False):
        print("It is a boolean value")
```

```
case _:  
    print("Not a boolean")
```

It is a boolean value

דוגמה למימוש משותף של משפט case ומשפט .if

```
n=int(input("Enter the num: "))  
match n:  
    case n if n < 0:  
        print("Number is negative")  
    case n if n == 0:  
        print("Number is zero")  
    case n if n > 0:  
        print("Number is positive")
```

Enter the num: -8  
Number is negative

## טיפול בשגיאות.

בכל תוכנית, גדולה במיוחד, עלולות להופיע שגיאות שגורמות לתוכנה לחוסר יכולת לפעול, ובסוף התוכנית לא עושה את מה שהיא צריכה לעשות. הגורמים לטעויות הם רבים. מתכנת יכול לטעות בשימוש בשפת התכנות עצמה.

במילים אחרות, כתיבת תוכנית יכולה ללוות בקשיים רבים, כשרוב הקשיים מתעסקים רק בטיפול בשגיאות, לכן זהו נושא גדול. לדוגמה, התחל את שם המשתנה במספר, או שכח להכניס מעי גם בכותרת של הוראות מורכבות. שגיאות כאלה נקראות תחביריות: הן מפרות את התחביר ואת פיסוק השפה. המתורגמן של פייתון, לאחר שנתקל בביטוי שגוי אינו יודע לפרש אותו. לכן הוא מפסיק את ביצוע התוכנית ומציג הודעה המתאימה המציינת את מיקום השגיאה.

```
>>> 1a=10  
SyntaxError: invalid syntax
```

בטרמינולוגיה של פייתון התרחשה כאן "חריגה" (Exception) ששייכת לקבוצה `SyntaxError`. על פי התיעוד, שגיאות תחביר בדרך כלל מיוחסות לטעויות, וכל השאר ל"חריגות". בכמה שפות תכנות לא משתמשים במילה "חריגה", ושגיאות מחולקות לתחביר וסמנטי. הפרת סמנטיקה פירושה בדרך כלל שלמרות שהביטויים מאויתים נכון מבחינת התחביר של השפה, התוכנית לא עובדת כמצופה. לשם השוואה ניתן לומר ממש כמה משפטים, אבל מבחינת המשמעות זה יהיה ללא הגיון, או שהמשתמשים לא יבינו כפי שרצו המפתחים. בפייתון לא מדובר בטעויות סמנטיות, אלא ב"חריגות", ויש רבים מהן. בשיעור זה נבדוק כמה מהם, ובעקבות זאת ניפגש עם עוד כמה. אם ננסה לגשת למשתנה שלא הוקצה לו ערך, שבמקרה של פייתון פירושו שהמשתנה לא הוכרז כלל ובעצם לא קיים, אז תזרוק חריגה של `NameError`.

```
>>> f=5
>>> print(f+w)
Traceback (most recent call last):
  File "<pyshell#46>", line 1, in <module>
    print(f+w)
NameError: name 'w' is not defined
```

בסעיף הקודם ראינו חריגה אחרת:

```
a=5>0
if a:
    print(a)
    b=7
    if b:
        print(b)
```

```
C:\Users\marka\AppData\Local\Programs\Python\Python38-32\python.exe C:/MyPython/third.py
File "C:/MyPython/third.py", line 6
    print(b)
    ^
IndentationError: expected an indented block
```

חריגות מסוג ValueError ו- TypeError.

חריגת ValueError קיימת כאשר פונקציה לא מסוגלת לחשב או להמיר ערך שקיבלה הפונקציה.

חריגת TypeError קיימת כאשר אנו מנסים לבצע פעולות בין סוגים שונים של המשתנים שלא מסוגלים לעבוד ביחד. (בעיקר בין מספרים לבין מחרוזות).

```
>>> int("Hi")
Traceback (most recent call last):
  File "<pyshell#61>", line 1, in <module>
    int("Hi")
ValueError: invalid literal for int() with base 10: 'Hi'
>>> 8+'3'
Traceback (most recent call last):
  File "<pyshell#62>", line 1, in <module>
    8+'3'
TypeError: unsupported operand type(s) for +: 'int' and 'str'
```

```
>>> 1/0
Traceback (most recent call last):
  File "<pyshell#63>", line 1, in <module>
    1/0
ZeroDivisionError: division by zero
```

## חריגה של פונקציות מתמטיות:

```
>>> import math
>>> math.sqrt(-1)
Traceback (most recent call last):
  File "<pyshell#65>", line 1, in <module>
    math.sqrt(-1)
ValueError: math domain error
>>> a=1
>>> math.sqrt(a)
1.0
>>> b=-1
>>> math.sqrt(b)
Traceback (most recent call last):
  File "<pyshell#69>", line 1, in <module>
    math.sqrt(b)
ValueError: math domain error
```

## טיפול בחריגות: try except (דומה למנגנון try and catch בשפת C#).

כאשר בתוכנית במהלך כתיבת הקוד או בדיקתו מתרחשות שגיאות, מתכנת יכול לתוקן את הקוד כך שלא יתרחשו שגיאות. עם זאת, לעתים קרובות פעולות המשתמש גורמות לחריגה בתוכנית. לדוגמה, התוכנית מצפה להזנת מספר, אך האדם הזין מכתב. ניסיון להמיר אותו למספר יזרוק חריגה של ValueError והתוכנית תתרוסק.

במקרה זה, בשפות תכנות, כולל Python, יש מפעיל מיוחד המאפשר לתפוס חריגים המתרחשים ולטפל בהם כך שהתוכנית תמשיך לעבוד או שתסיים את עבודתה כראוי. בפיתוח, יירוט כזה מבוצע על ידי הניסיון מלבד אמירה. "try" מתורגם ל"נסה", "except" - כחריגה. אפשר לתאר את עבודתו כך: "נסה לעשות זאת וכי אם מתרחש חריגה, עשה זאת ואחרת."

כיצד זה עובד?

קודם כל מבצעים פעולות של סעיף "try" (זאת אומרת מריצים קוד שנמצא בין try לבין except).

אם אין שגיאות ("יוצא מן הכלל") אז רק "try" מתבצע ו- "יוצא מן הכלל" לא מופעל.

אחרת (במקרה של שגיאה כל שהיא) מתבצע "יוצא מן הכלל".



לכל משפט try ניתן לחבר מספר משפטים except – כל אחד עם שגיאה משלו.  
לדוגמה:

```
n = input("Enter integer: ")
try:
    n = int(n)
    print("Really Integer")
except:
    print("Not Integer")
```

```
Enter integer: as
Not Integer
Enter integer: 12.1
Not Integer
Enter integer: 151
Really Integer
```

אם ישנם מספר שגיאות, עדיף לטפל בכל סוג שגיאה בנפרד:

```
try:
    a = float(input("Enter first number: "))
    b = float(input("Enter second number: "))
    c = a / b
    print("a/b = %.2f" % c)
except ValueError:
    print("You cant enter string")
except ZeroDivisionError:
    print("You cant divide by zero")
```

```
Enter first number: 12
Enter second number: 2.5
a/b = 4.80

Enter first number: 12
Enter second number: n
You cant enter string

Enter first number: 3
Enter second number: 0
You cant divide by zero
```

## פקודות נוספות: " else ", " finally " .

```
try:  
    n = input('Enter the integer: ')  
    n = int(n)  
except ValueError:  
    print("Something wrong with the input")  
else: # חלק זה בוצע כאשר לא הייתה חריגה  
    print("Input is right", n)  
finally: # מתבצע בכל זאת  
    print("End of program")
```

```
Enter the integer: 1.2  
Something wrong with the input  
End of program
```

```
Enter the integer: hkjh  
Something wrong with the input  
End of program
```

```
Enter the integer: 121  
Input is right 121  
End of program
```

ישנה רשימה של כל ה-exceptions. שימו לב שם מסודרות לפי רמות חשיבות:

```
BaseException
+-- SystemExit
+-- KeyboardInterrupt
+-- GeneratorExit
+-- Exception
  +-- StopIteration
  +-- StopAsyncIteration
  +-- ArithmeticError
  |   +-- FloatingPointError
  |   +-- OverflowError
  |   +-- ZeroDivisionError
  +-- AssertionError
  +-- AttributeError
  +-- BufferError
  +-- EOFError
  +-- ImportError
  |   +-- ModuleNotFoundError
  +-- LookupError
  |   +-- IndexError
  |   +-- KeyError
  +-- MemoryError
  +-- NameError
  |   +-- UnboundLocalError
  +-- OSError
  |   +-- BlockingIOError
  |   +-- ChildProcessError
  |   +-- ConnectionError
  |   |   +-- BrokenPipeError
  |   |   +-- ConnectionAbortedError
  |   |   +-- ConnectionRefusedError
  |   |   +-- ConnectionResetError
  |   +-- FileExistsError
  |   +-- FileNotFoundError
  |   +-- InterruptedError
  |   +-- IsADirectoryError
  |   +-- NotADirectoryError
  |   +-- PermissionError
  |   +-- ProcessLookupError
  |   +-- TimeoutError
  +-- ReferenceError
  +-- RuntimeError
  |   +-- NotImplementedError
  |   +-- RecursionError
  +-- SyntaxError
  |   +-- IndentationError
  |   +-- TabError
  +-- SystemError
  +-- TypeError
  +-- ValueError
  |   +-- UnicodeError
  |   |   +-- UnicodeDecodeError
  |   |   +-- UnicodeEncodeError
  |   |   +-- UnicodeTranslateError
  +-- Warning
  |   +-- DeprecationWarning
  |   +-- PendingDeprecationWarning
  |   +-- RuntimeWarning
  |   +-- SyntaxWarning
  |   +-- UserWarning
  |   +-- FutureWarning
  |   +-- ImportWarning
  |   +-- UnicodeWarning
  |   +-- BytesWarning
  |   +-- ResourceWarning
```

ישנן אפשרויות לתת "יוצא מן הכלל" באופן כללי או אפשר לדייק – על איזה סוג השגיאה מדובר. לדוגמא בבעיות מתמטיות ArithmeticError יש 3 סוגי exceptions : OverflowError, FloatingPointError, ZeroDivisionError.

```
try:
    numerator=int(input(" Numerator = "))
    denominator=int(input(" Denominator = "))
    print(f'In the try block: {numerator}/{denominator}')
    result = numerator / denominator
except ZeroDivisionError as zde:
    print(zde)
else:
    print('The result is:', result)
finally:
    print('Exiting')
```

```
Numerator = 6
Denominator = 0
In the try block: 6/0
division by zero
Exiting
```

תרגול:

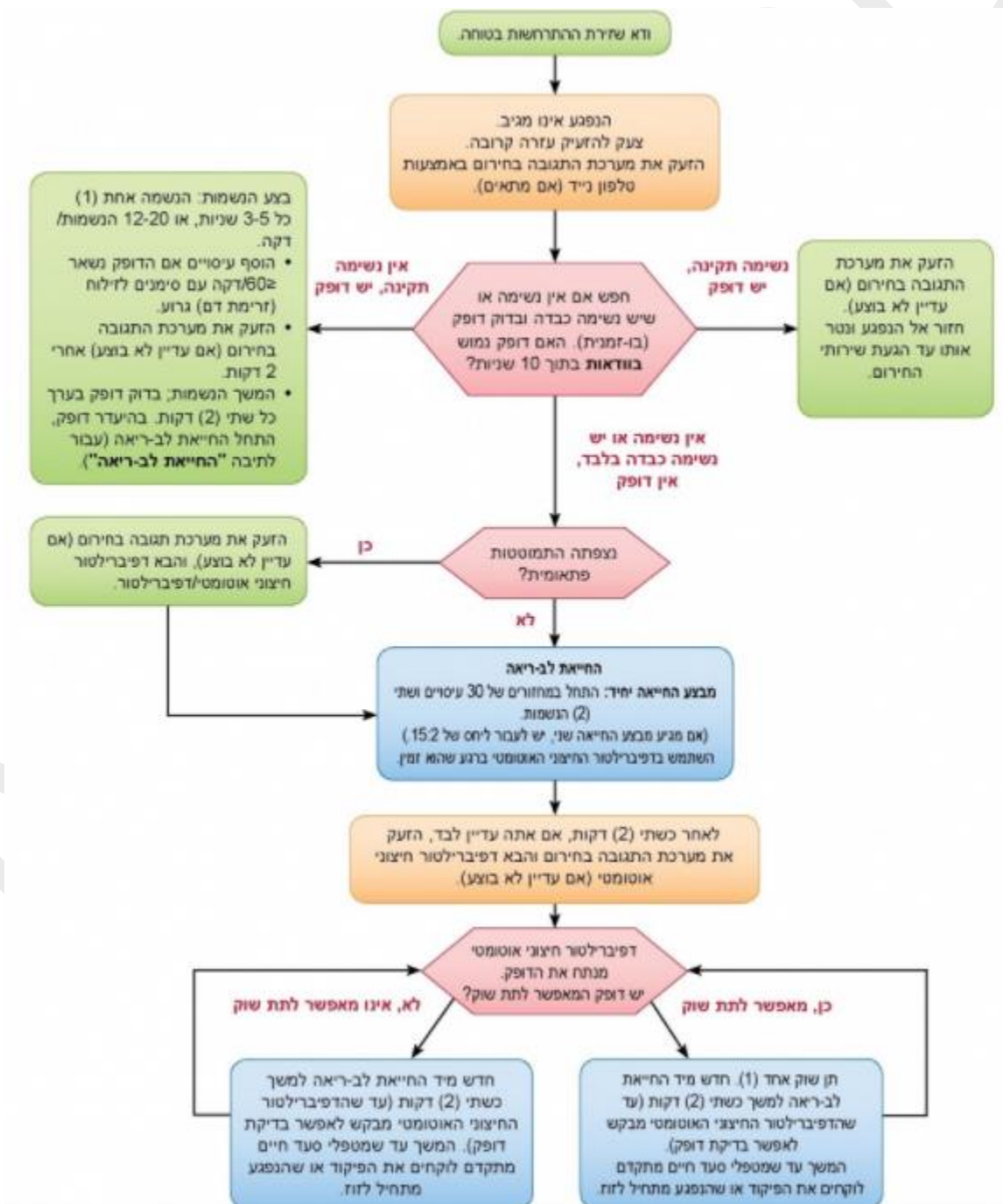
כתוב תכנית הקולטת מהמשתמש שני נתונים. אם שניהם מספרים, התוכנית תחבר אותם, אם לפחות אחד מהם מסוג מחרוזת, התוכנית תבנה משתי נתונים האלה מחרוזת אחידה. ל דוגמה:

```
First: 4
Second: 5
Result: 9.0
```

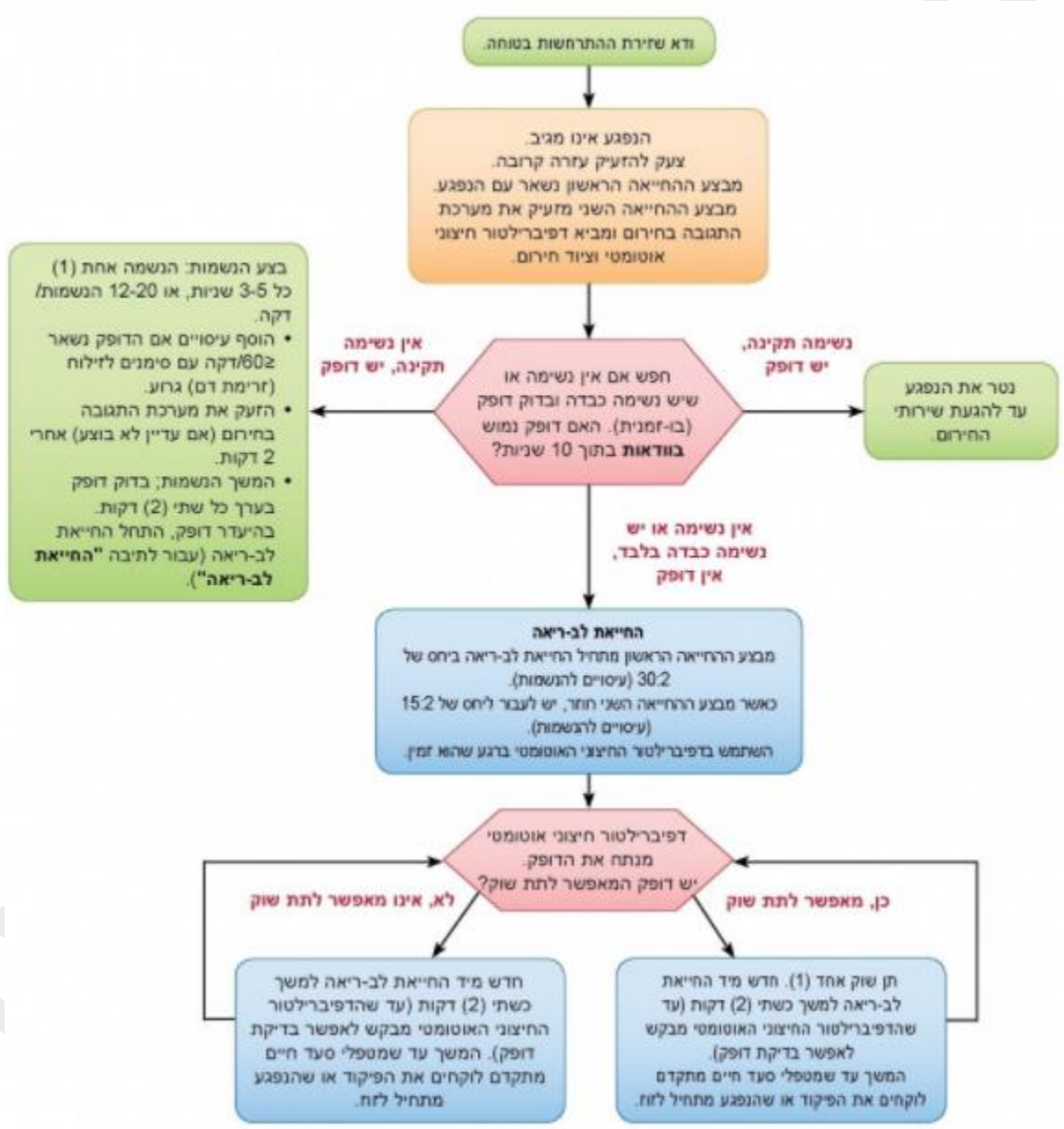
```
First: a
Second: 5
Result: a5
```

# תרגול בנושא ביצוע מותנה: תרגול בנושא תרשימי זרימה. תרגיל מס' 1.

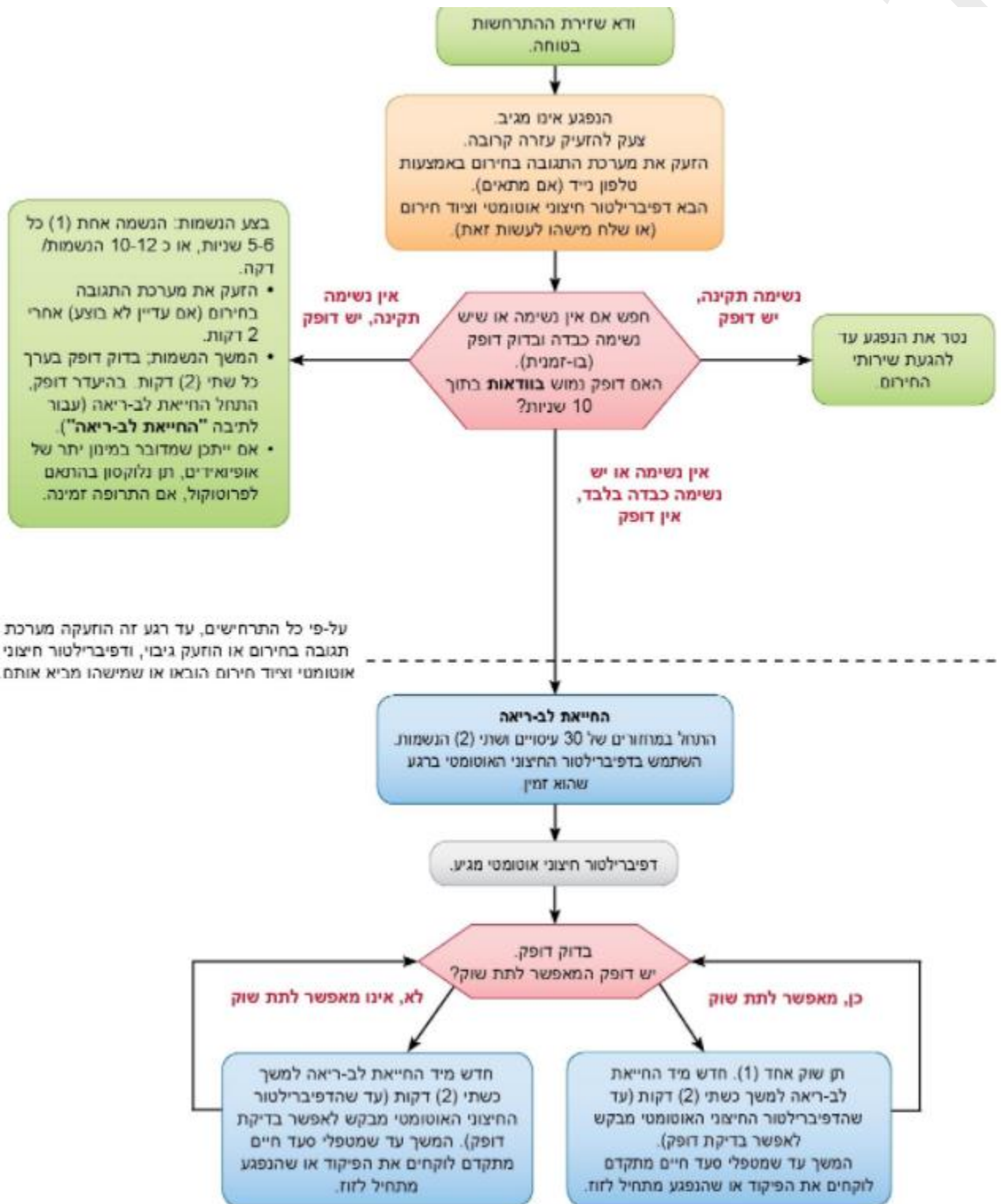
תעבור על אלגוריתם של החייאה מדום לב ע"י אדם אחד ותקן את הטעויות וצייר אותו מחדש בהתאם לחוקים של הנדסת תוכנה.



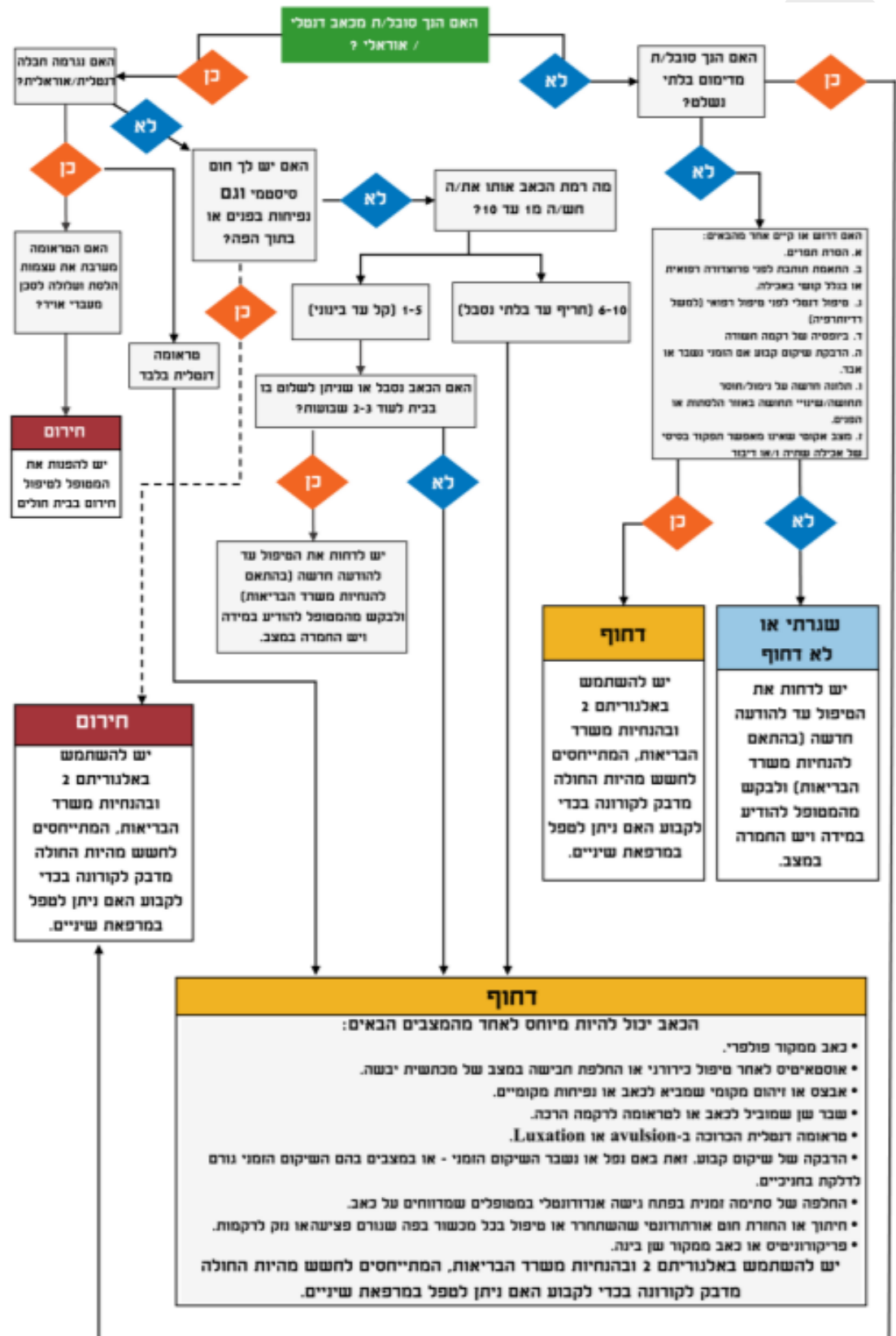
תעבור על אלגוריתם של החייאה מדום לב ע"י אדם אחד ותקן את הטעויות וצייר אותו מחדש בהתאם לחוקים של הנדסת תוכנה.



תעבור על אלגוריתם של החייאה מדום לב ע"י אדם אחד ותקן את הטעויות וצייר אותו מחדש בהתאם לחוקים של הנדסת תוכנה.



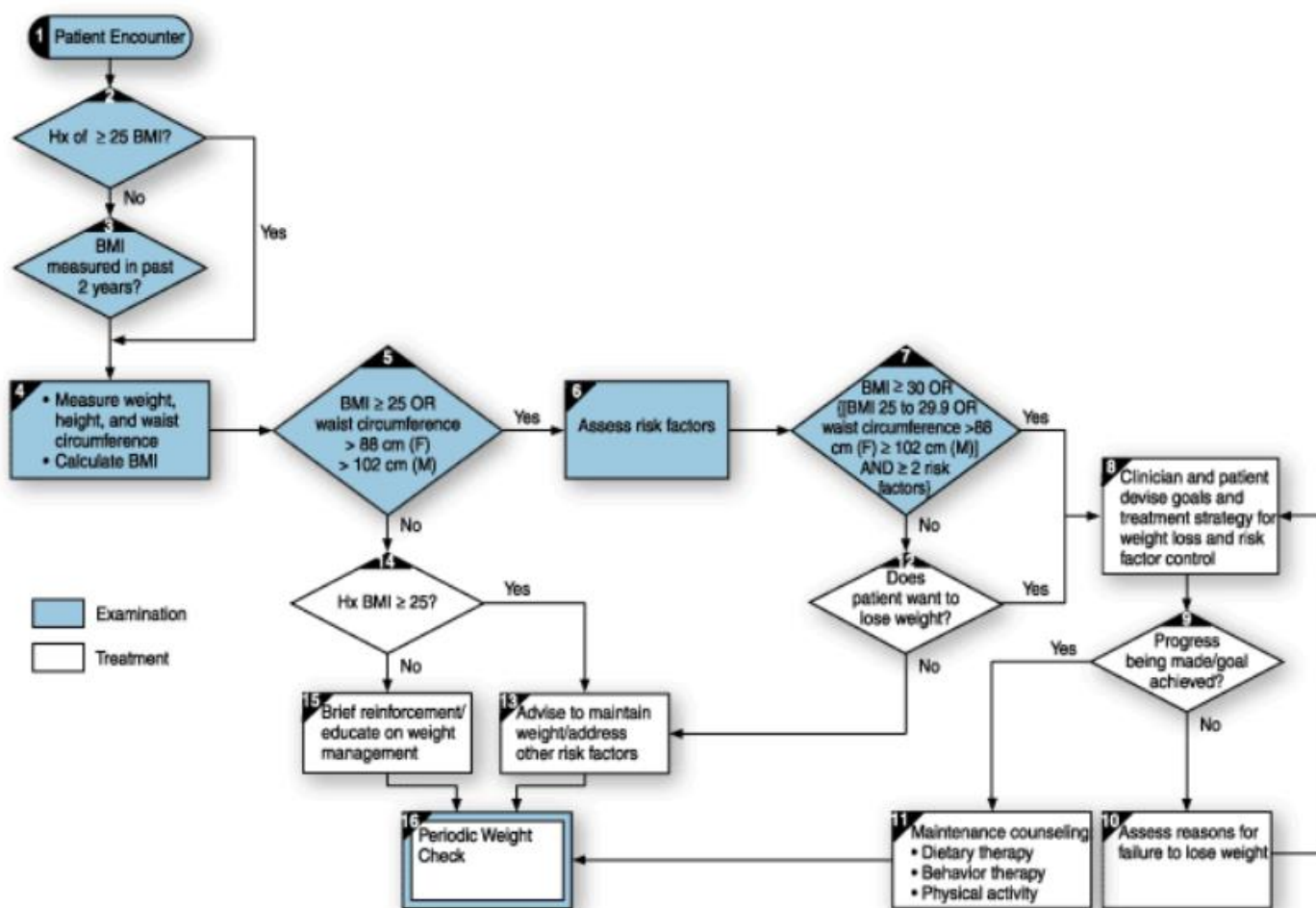
תעבור על האלגוריתם של טיפול בכאבי שיניים ותיייר אותו מחדש בהתאם לחוקים של הנדסת תוכנה.





תרגיל מס' 5.

נתון אלגוריתם לגילוי וטיפול בעודף משקל:



תעבור על אלגוריתם של גילוי וטיפול בעודף משקל ותצייר אותו מחדש בהתאם לחוקים של הנדסת תוכנה.

רשום הסבר מילולי לאלגוריתם הנתון.

תן מספר דוגמאות וציין דרך גילוי וטיפול לכל מקרה לפי אלגוריתם הנתון.

תרגיל מס' 6.

כתוב תרשימי זרימה לתוכנית הבאה:

קולטים מהמשתמש 3 זוויות של משולש. במידה ומשולש חוקי (סכום זוויות 180 מעלות) בדוק האם משולש ישר זווית, כהה זווית או רגיל. בנוסף תבדוק האם הוא שוויון שוקיים או שוויון צלעות. שים לב שמשולש יכול להיות גם שוויון שוקיים וגם ישר זווית או כהה זווית.

תרגיל מס' 7.

כתוב תרשימי זרימה לתוכנית הבאה:

קבל מהמשתמש מספר כלשהו שלם חיובי. תבדוק האם הוא מתחלק ב-2, 3, 9? (מספר שמסתיים ב-0, 2, 4, 6, 8 מתחלק ב-2, מספר שסכום הספרות שלו מתחלק ב-3 גם מתחלק ב-3, מספר שסכום הספרות שלו מתחלק ב-9 גם מתחלק ב-9).

תרגיל מס' 8.

כתוב תרשימי זרימה לתוכנית הבאה:

קבל מהמשתמש מספר כלשהו שלם חיובי. תבדוק האם הוא מתחלק ב-2, 3, 5, 9, 10? (מספר שמסתיים ב-0, 2, 4, 6, 8 מתחלק ב-2, מספר שסכום הספרות שלו מתחלק ב-3 גם מתחלק ב-3, מספר שסכום הספרות שלו מתחלק ב-9 גם מתחלק ב-9, אם מספר מסתיים ב-0 או ב-5 הוא מתחלק ב-5 ואם הוא מסתיים ב-0 הוא מתחלק ב-10).

תרגיל מס' 9.

כתוב תרשימי זרימה לתוכנית הבאה:

תוכנית מקבלת מהמשתמש 4 צלעות ו-4 זוויות של מרובה ובודקת האם זה מלבן, ריבוע, מקבילית, מעוין או טרפז.

תרגיל מס' 10.

כתוב תרשימי זרימה לתוכנית הבאה:

קולטים שני מספרים שלמים וחיוביים ומוציאים מחנה משותף הגדול ביותר.

תרגיל מס' 11.

כתוב תרשימי זרימה לתוכנית הבאה:

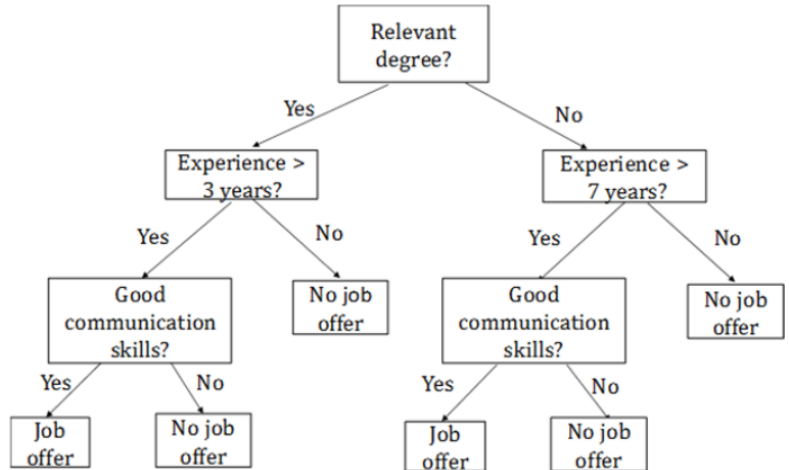
קולטים שני מספרים שלמים וחיוביים ומוציאים מחלק משותף הקטן ביותר (גדול מ-1).

תרגיל מס' 12.

כתוב תרשימי זרימה לחיפוש שורש ריבועי ממספר חיובי שלם בשיטה היעילה ביותר (שיטת חצייה).

תרגיל מס' 13.

נתון אלגוריתם לבחירת מועמד לעבודה.



פשט אותו וצייר אותו מחדש לפי כללים של הנדסת תוכנה.

תרגול בנושא טבלאות מעקב.

תרגיל מס' 1.

תפענח את הקוד הבא והצג מספר דוגמאות של פלט עבור קלט שונה.

```
x=float(input("Enter the X"))
if x>0:
    y=x-0.5
elif x==0:
    y=0
else:
    y=-x
print(y)
```

תרגיל מס' 2.

תפענח את הקוד הבא והצג מספר דוגמאות של פלט עבור קלט שונה.

```
ch=int(input("What You want to know? Mass - 1, Volume - 2 Density - 3"))
if ch==1:
    v=float(input("Enter the volume "))
    ro=float(input("Enter the density "))
    m=v*ro
elif ch==2:
    m = float(input("Enter the mass "))
    ro = float(input("Enter the density "))
    v=m/ro
else:
```

```

v = float(input("Enter the volume "))
m = float(input("Enter the mass "))
ro=m/v
print("Mass = " + str(m) + " Denisty = " + str(ro) + " Volume " + str(v) )

```

תרגיל מס' 3.

תפענח את הקוד הבא והצג מספר דוגמאות של פלט עבור קלט שונה.

```

import math
ch=int(input("What figure are You talking about? Rectangle - 1, Circle - 2, Triangle - 3 "))
if ch==1:
    a=float(input("Enter first side "))
    b=float(input("Enter second side "))
    S=a*b
elif ch==2:
    r = float(input("Enter the radius "))
    S=math.pi*r*r
else:
    a = float(input("Enter first side "))
    b = float(input("Enter second side "))
    c = float(input("Enter third side "))
    p=(a+b+c)/2
    S=math.sqrt(p*(p-a)*(p-b)*(p-c))
print("Square is " + str(S))

```

תרגיל מס' 4.

תפענח את הקוד הבא והצג מספר דוגמאות של פלט עבור קלט שונה.

```

import math
x = float(input("Enter x "))
y = float(input("Enter y "))
r = float(input("Enter radius "))
c=math.sqrt(x*x+y*y)
if c==r:
    print("The point is on the circle")
elif c>r:
    print("The point is out of the circle")
else:
    print("The point is in the circle")

```

תרגיל מס' 5.

תפענח את הקוד הבא והצג מספר דוגמאות של פלט עבור קלט שונה.

```
x = float(input("Enter x "))
y = float(input("Enter y "))
r = input("Enter operator ")
c=0
if r=='+' :
    c=x+y
elif r=='-' :
    c=x-y
elif r=='*' :
    c=x*y
elif r=='/' :
    c=x/y
else:
    print("Unknown function")
print(c)
```

תרגיל מס' 6.

תפענח את הקוד הבא והצג מספר דוגמאות של פלט עבור קלט שונה.

```
flag = input(" What You want to check? (m, d, v): ")
result = 0
if flag == 'm':
    d = float(input(" Density "))
    v = float(input(" Volume "))
    result = d * v
elif flag == 'd':
    m = float(input(" Mass "))
    v = float(input(" Volume "))
    result = m / v
elif flag == 'v':
    m = float(input(" Mass "))
    d = float(input(" Density "))
    result = m / d
print(" %.2f" % result)
```

תרגיל מס' 7.

תפענח את הקוד הבא והצג מספר דוגמאות של פלט עבור קלט שונה.

```
a = float(input("Enter first side "))
b = float(input("Enter second side "))
```

```
c = float(input("Enter third side "))
if a<b+c and b<a+c and c<a+b:
    print("Triangle is legal")
else:
    print("Triangle is not legal")
```

תרגיל מס' 8.

תפענח את הקוד הבא והצג מספר דוגמאות של פלט עבור קלט שונה.

```
print("Enter x and y:")
x = float(input("x = "))
y = float(input("y = "))
if x == 0 and y == 0:
    print("At the center")
elif x == 0:
    print("Axis 'x'")
elif y == 0:
    print("Axis 'y'")
elif x > 0:
    if y>0:
        print(" I quarter")
    elif y < 0:
        print("IV quarter")
elif x < 0:
    if y>0:
        print("II quarter")
    elif y<0:
        print("III quarter")
```

תרגיל מס' 9.

תפענח את הקוד הבא והצג מספר דוגמאות של פלט עבור קלט שונה.

```
year=int(input("Please enter thd year "))
if year%1000==0:
    print("Leap Year")
elif year%100==0:
    print("Not Leap Year")
elif year%4==0:
    print("Leap Year")
else:
    print("Not Leap Year")
```

תרגיל מס' 10.

תפענח את הקוד הבא והצג מספר דוגמאות של פלט עבור קלט שונה.

```
temper=int(input("Please enter the temperature "))
tempt=input("Please enter type of temperature F or C ")
if(tempt=="F" or tempt=="f"):
    tempC=float((temper-32)*5/9)
    print(tempC)
elif (tempt=="C" or tempt=="c"):
    tempF=float(temper*9/5+32)
    print(tempF)
else:
    print("Unknown letter")
```

תרגיל מס' 11.

תפענח את הקוד הבא והצג מספר דוגמאות של פלט עבור קלט שונה.

```
import random
A = random.randrange(-3,3)
B = random.randrange(-3,3)
C = random.randrange(-3,3)
print("Three Numbers:", A, B, C)
x = 0
if A > 0:
    x += 1
if B > 0:
    x += 1
if C > 0:
    x += 1
print("The number of positive : ", x)
```

תרגיל מס' 12.

תפענח את הקוד הבא והצג מספר דוגמאות של פלט עבור קלט שונה.

```
print("1-rectangle, 2-triangle, 3-circle")
figure = input("Choose one of them: ")
if figure == '1':
    print("Enter sides of rectangle:")
    a = float(input("a = "))
    b = float(input("b = "))
    print("Square is: %.2f" % (a * b))
elif figure == '2':
    print("Enter 3 sides of triangle:")
```

```

a = float(input("a = "))
b = float(input("b = "))
c = float(input("c = "))
p = (a + b + c) / 2
from math import sqrt
s = sqrt(p * (p - a) * (p - b) * (p - c))
print("Square is: %.2f" % s)
elif figure == '3':
    r = float(input("Radius of the circle R = "))
    from math import pi
    print("Square is: %.2f" % (pi * r ** 2))
else:
    print("Error of input")

```

תרגיל מס' 13.

תפענח את הקוד הבא והצג מספר דוגמאות של פלט עבור קלט שונה.

```

print("Length of sides:")
a = float(input("a = "))
b = float(input("b = "))
c = float(input("c = "))
flag = ""
if a + b > c:
    if a + c > b:
        if b + c > a:
            print("Triangle ")
        else:
            flag = 'a'
    else:
        flag = 'b'
else:
    flag = 'c'
if flag != "":
    print("Not triangle")
    print("%s' > more than sum" % flag)

```

תרגיל מס' 14.

תפענח את הקוד הבא והצג מספר דוגמאות של פלט עבור קלט שונה.

```

s = input("Enter the sign (+,-,*,/): ")
if s in ('+', '-', '*', '/'):
    x = float(input("x="))
    y = float(input("y="))

```



```

if s == '+':
    print("%.2f" % (x+y))
elif s == '-':
    print("%.2f" % (x-y))
elif s == '*':
    print("%.2f" % (x*y))
elif s == '/':
    if y != 0:
        print("%.2f" % (x/y))
    else:
        print("You cant divide by zero!")
else:
    print("Illegal sign")

```

תרגיל מס' 15.

תפענח את הקוד הבא והצג מספר דוגמאות של פלט עבור קלט שונה.

```

x=int(input("Please enter x "))
y=int(input("Please enter y "))
if x>0 and y>0:
    print("Your point in 1 quarter")
elif x<0 and y>0:
    print("Your point in 2 quarter")
elif x<0 and y<0:
    print("Your point in 3 quarter")
elif x>0 and y<0:
    print("Your point in 4 quarter")
else:
    print("Your point is on the line")

```

תרגיל מס' 16.

תפענח את הקוד הבא והצג מספר דוגמאות של פלט עבור קלט שונה.

```

year = int(input())
if year % 4 != 0:
    print("Regular")
elif year % 100 == 0:
    if year % 400 == 0:
        print("Leap")
    else:
        print("Regular")
else:
    print("Leap")

```

תפענח את הקוד הבא והצג מספר דוגמאות של פלט עבור קלט שונה.

```
import math
print("Now we shall solve  $ax^2+bx+c=0$ ")
a=int(input("Please enter a: "))
b=int(input("Please enter b: "))
c=int(input("Please enter c: "))
if a==0:
    if b==0:
        if c!=0:
            print("No roots")
        else:
            print("Infinite roots")
    else:
        x=-c/b
        print("x= " + str(x))
elif b==0:
    if -c/a>0:
        x1= math.sqrt(-c/a)
        x2=-x1
        print(" x1= " + str(x1)+ " x2= " + str(x2))
    else:
        print("Complex roots")
elif c==0:
    x1 = 0
    x2 = -b/a
    print(" x1= " + str(x1) + " x2= " + str(x2))
else:
    D=b*b-4*a*c
    if D<0:
        print("Complex roots")
    elif D==0:
        x=-b/(2*a)
        print("x= " + str(x))
    else:
        x1=(-b+math.sqrt(D))/(2*a)
        x2=(-b-math.sqrt(D))/(2*a)
        print(" x1= " + str(x1) + " x2= " + str(x2))
```

תרגיל מס' 18.

תפענח את הקוד הבא והצג מספר דוגמאות של פלט עבור קלט שונה.

```
a=int(input("Please enter amount in kW/h "))
if a<=100:
    sum=a*0.1
elif a<=200:
    sum=100*0.1+(a-100)*0.2
elif a<300:
    sum = 100 * 0.1 + 100*0.2 + (a - 200) * 0.3
else:
    sum = 100 * 0.1 + 100 * 0.2 + 100*0.3 + (a - 300) * 0.4
print("You had to pay: " + str(sum) + " shekel ")
```

תרגיל מס' 19.

תפענח את הקוד הבא והצג מספר דוגמאות של פלט עבור קלט שונה.

```
import math
a=int(input("Please enter the Numerator "))
b=int(input("Please enter the Denominator "))
try:
    c=a/b
    print("a/b= " + str(c))
    try:
        d = math.log(c, 10)
        print("log(a/b) = " + str(d))
    except:
        print("You cant make log on negative number")
except:
    print("You cant divide by zero")
```

תרגיל מס' 20.

תפענח את הקוד הבא והצג מספר דוגמאות של פלט עבור קלט שונה.

```
try:
    let = int(input("Enter the number of the letter: "))
    if let>=1 and let <=26:
        letter=chr(65+let-1)
        print(letter)
    else:
        print("invalid number")
except ValueError:
    print(" You havent enter integer number")
```

תרגיל מס' 21.

תפענח את הקוד הבא והצג מספר דוגמאות של פלט עבור קלט שונה.

```
import random
a=int(input("Enter minimum number: "))
b=int(input("Enter maximum number: "))
if a<=b:
    c=random.randint(a,b)
else:
    c = random.randint(b, a)
print("Random number is " + str(c))
```

תרגיל מס' 22.

תפענח את הקוד הבא והצג מספר דוגמאות של פלט עבור קלט שונה.

```
import random
A = random.randrange(-3,3)
B = random.randrange(-3,3)
C = random.randrange(-3,3)
print("Three numbers:", A, B, C)
x = 0
if A > 0:
    x += 1
if B > 0:
    x += 1
if C > 0:
    x += 1
print("Positive: ", x)
y = 0
if A < 0:
    y += 1
if B < 0:
    y += 1
if C < 0:
    y += 1
print("Negative: ", y)
```

תרגיל מס' 23.

תפענח את הקוד הבא והצג מספר דוגמאות של פלט עבור קלט שונה.

```
A = random.randrange(-3,3)
B = random.randrange(-3,3)
print("Two Numbers:", A, B)
```

```
if A > B:  
    print(A,B)  
else:  
    print(B,A)
```

תרגיל מס' 24.

תפענח את הקוד הבא והצג מספר דוגמאות של פלט עבור קלט שונה.

```
import random  
A = random.randrange(-3,3)  
B = random.randrange(-3,3)  
print("Number A:", A)  
print("Number B:", B)  
if A != B:  
    A = B = max(A,B)  
else:  
    A = B = 0  
print("Number A:", A)  
print("Number B:", B)
```

תרגיל מס' 25.

תפענח את הקוד הבא והצג מספר דוגמאות של פלט עבור קלט שונה.

```
import random  
A = random.randrange(-30,30)  
B = random.randrange(-30,30)  
C = random.randrange(-30,30)  
print("Number A:", A)  
print("Number B:", B)  
print("Number C:", C)  
if (A <= B and B <= C) or (C <= B and B <= A):  
    m = B  
elif (B <= A and A <= C) or (C <= A and A <= B):  
    m = A  
elif (A <= C and C <= B) or (B <= C and C <= A):  
    m = C  
print()  
print("Middle:", m)
```

תפענח את הקוד הבא והצג מספר דוגמאות של פלט עבור קלט שונה.

```
import random
A = random.randrange(-30,30)
B = random.randrange(-30,30)
C = random.randrange(-30,30)
print("Number A:", A)
print("Number B:", B)
print("Number C:", C)
if (A < B and B < C) or (A > B and B > C):
    k = 2
else:
    k = -1
A *= k
B *= k
C *= k
print()
print("Number A:", A)
print("Number B:", B)
print("Number C:", C)
```

תפענח את הקוד הבא והצג מספר דוגמאות של פלט עבור קלט שונה.

```
import random
A = random.randrange(-30,30)
B = random.randrange(-30,30)
C = random.randrange(-30,30)
print("Number A:", A)
print("Number B:", B)
print("Number C:", C)
if A < B:
    mn = A
else:
    mn = B
if mn > C:
    mn = C
print()
print("Minimum:", mn)
```

תפענח את הקוד הבא והצג מספר דוגמאות של פלט עבור קלט שונה.

```
import random
A = random.randrange(-30,30)
B = random.randrange(-30,30)
C = random.randrange(-30,30)

print("Number A:", A)
print("Number B:", B)
print("Number C:", C)
if A < B:
    mn = A
    mx = B
else:
    mn = B
    mx = A
if mx < C:
    mx = C
if mn > C:
    mn = C
print()
print("Minimum:", mn)
print("Maximum:", mx)
```

תפענח את הקוד הבא והצג מספר דוגמאות של פלט עבור קלט שונה.

```
import random
A = random.randrange(-3,3)
B = random.randrange(-3,3)
print("Number A:", A)
print("Number B:", B)
if A > B:
    A,B = B,A
print("Number A:", A)
print("Number B:", B)
```

תרגיל מס' 30.

תפענח את הקוד הבא והצג מספר דוגמאות של פלט עבור קלט שונה.

```
A = random.randrange(-30,30)
B = random.randrange(-30,30)
C = random.randrange(-30,30)
print("Number A:", A)
print("Number B:", B)
print("Number C:", C)
if (C <= A and A <= B) or (C <= B and B <= A):
    m = A + B
elif (B <= A and A <= C) or (B <= C and C <= A):
    m = A + C
elif (A <= C and C <= B) or (A <= B and B <= C):
    m = B + C
print()
print("Sum is:", m)
```

תרגיל מס' 31.

תפענח את הקוד הבא והצג מספר דוגמאות של פלט עבור קלט שונה.

```
import random
A,B,C = random.sample(range(-10, 10), 3)
print("Number A:", A)
print("Number B:", B)
print("Number C:", C)
AB = abs(A-B)
AC = abs(A-C)
print("AB:",AB)
print("AC:",AC)
if AB < AC:
    print("B is closer to A")
elif AB > AC:
    print("C is closer to A")
else:
    print("A is in the center of BC")
```



תפענח את הקוד הבא והצג מספר דוגמאות של פלט עבור קלט שונה.

```
import random
A = random.randrange(-30,30)
B = random.randrange(-30,30)
C = random.randrange(-30,30)
print("Number A:", A)
print("Number B:", B)
print("Number C:", C)
if (A < B and B < C):
    k = 2
else:
    k = -1
A *= k
B *= k
C *= k
print()
print("Number A:", A)
print("Number B:", B)
print("Number C:", C)
```

תפענח את הקוד הבא והצג מספר דוגמאות של פלט עבור קלט שונה.

```
import random
A = random.randrange(-3,3)
B = random.randrange(-3,3)
print("Number A:", A)
print("Number B:", B)
if A != B:
    A = B = A+B
else:
    A = B = 0
print("Number A:", A)
print("Number B:", B)
```

תפענח את הקוד הבא והצג מספר דוגמאות של פלט עבור קלט שונה.

```
import math
A=input("Enter value convert from: a for catetus, h for height, c for hypotenuse, s for square: ")
val=float(input("Enter the value: "))
match A:
    case 'a'|'A':
        print(" height is = "+str(val/math.sqrt(2)))
        print(" hypotenuse = " + str(val*math.sqrt(2)))
        print(" square = " + str(val**2/2))
    case 'h'|'H':
        print(" catetus = " + str(val *math.sqrt(2)))
        print(" hypotenuse = " + str(2*val))
        print(" square = " + str(val**2))
    case 'c'|'C':
        print(" catetus = " + str(val / math.sqrt(2)))
        print(" height = " + str(val /2))
        print(" square = " + str(val ** 2 /4))
    case 's'|'S':
        print(" catetus = " + str(math.sqrt(val*2)))
        print(" height = " + str(math.sqrt(val)))
        print(" hypotenuse = " + str(math.sqrt(val)*2))
    case _:
        print("No Such operator")
```

תפענח את הקוד הבא והצג מספר דוגמאות של פלט עבור קלט שונה.

```
A=int(input("Enter the month of birthday: "))
B=int(input("Enter the date of birthday: "))
match A:
    case 1:
        if 0<B<=19:
            print("You are Capricornus")
        elif B<=31:
            print("You are Aquarius")
        else:
            print("No Such Day")
    case 2:
        if 0 < B <= 18:
            print("You are Aquarius")
        elif B <= 28:
            print("You are Pisces")
```

```

else:
    print("No Such Day")
case 3:
    if 0 < B <= 18:
        print("You are Pisces")
    elif B <= 31:
        print("You are Aries")
    else:
        print("No Such Day")
case 4:
    if 0 < B <= 19:
        print("You are Aries")
    elif B <= 30:
        print("You are Taurus")
    else:
        print("No Such Day")
case 5:
    if 0 < B <= 20:
        print("You are Taurus")
    elif B <= 31:
        print("You are Gemini")
    else:
        print("No Such Day")
case 6:
    if 0 < B <= 21:
        print("You are Gemini")
    elif B <= 30:
        print("You are Cancer")
    else:
        print("No Such Day")
case 7:
    if 0 < B <= 22:
        print("You are Cancer")
    elif B <= 31:
        print("You are Leo")
    else:
        print("No Such Day")
case 8:
    if 0 < B <= 22:
        print("You are Leo")
    elif B <= 31:
        print("You are Virgo")
    else:
        print("No Such Day")

```

```

case 9:
    if 0 < B <= 22:
        print("You are Virgo")
    elif B <= 30:
        print("You are Libra")
    else:
        print("No Such Day")
case 10:
    if 0 < B <= 23:
        print("You are Libra")
    elif B <= 31:
        print("You are Scorpius")
    else:
        print("No Such Day")
case 11:
    if 0 < B <= 21:
        print("You are Scorpius")
    elif B <= 30:
        print("You are Sagittarius")
    else:
        print("No Such Day")
case 12:
    if 0 < B <= 21:
        print("You are Sagittarius")
    elif B <= 28:
        print("You are Capricornus")
    else:
        print("No Such Day")
case _:
    print("No Such Month")

```

תרגיל מס' 36.

תפענח את הקוד הבא והצג מספר דוגמאות של פלט עבור קלט שונה.

```

A=int(input("Enter the year of birthday after 1920: "))
A-=1920
A%=12
match A:
    case 0:
        print("Monkey")
    case 1:
        print("Rooster")
    case 2:
        print("Dog")

```

```

case 3:
    print("Pig")
case 4:
    print("Rat")
case 5:
    print("Ox")
case 6:
    print("Tiger")
case 7:
    print("Rabbit")
case 8:
    print("Dragon")
case 9:
    print("Snake")
case 10:
    print("Horse")
case 11:
    print("Goat")

```

תרגיל מס' 37.

תפענח את הקוד הבא והצג מספר דוגמאות של פלט עבור קלט שונה.

```

year = int(input())
if year % 4 != 0 or (year % 100 == 0 and year % 400 != 0):
    print("Regular")
else:
    print("Leap")

```

תרגיל מס' 38.

תפענח את הקוד הבא והצג מספר דוגמאות של פלט עבור קלט שונה.

```

import math
print("Now we shall solve ax2+bx+c=0")
a=int(input("Please enter a: "))
b=int(input("Please enter b: "))
c=int(input("Please enter c: "))
D=b*b-4*a*c
if D<0:
    print("Complex roots")
elif D==0:
    x=-b/(2*a)
    print("x= " + str(x))
else:

```

```
x1=(-b+math.sqrt(D))/(2*a)
x2=(-b-math.sqrt(D))/(2*a)
print(" x1= " + str(x1) + " x2= " + str(x2))
```

תרגיל מס' 39.

תפענח את הקוד הבא והצג מספר דוגמאות של פלט עבור קלט שונה.

```
n=int(input("Enter the num: "))
match n:
    case 1:
        print("Sunday")
    case 2:
        print("Monday")
    case 3:
        print("Tuesday")
    case 4:
        print("Wednesday")
    case 5:
        print("Thursday")
    case 6:
        print("Friday")
    case 7:
        print("Saturday")
    case _:
        print("No Such Day")
```

תרגיל מס' 40.

תפענח את הקוד הבא והצג מספר דוגמאות של פלט עבור קלט שונה.

```
n=int(input("Enter the mark from 1 to 10: "))
match n:
    case 1|2|3:
        print("Troll")
    case 4|5:
        print("Bad mark")
    case 6|7:
        print("Not Bad")
    case 8|9:
        print("Good")
    case 10:
        print("Excellent")
    case _:
        print("No Such mark")
```

תרגיל מס' 41.

תפענח את הקוד הבא והצג מספר דוגמאות של פלט עבור קלט שונה.

```
n=int(input("Enter the month from 1 to 12: "))
match n:
    case 12|1|2:
        print("Winter")
    case 3|4|5:
        print("Spring")
    case 6|7|8:
        print("Summer")
    case 9|10|11:
        print("Autumn")
    case _:
        print("No Such month")
```

תרגיל מס' 42.

תפענח את הקוד הבא והצג מספר דוגמאות של פלט עבור קלט שונה.

```
n=int(input("Enter the month from 1 to 12: "))
match n:
    case 1|3|5|7|8|10|12:
        print("31 days")
    case 4|6|9|11:
        print("30 days")
    case 2:
        print("28-29 Days")
    case _:
        print("No Such month")
```

תרגיל מס' 43.

תפענח את הקוד הבא והצג מספר דוגמאות של פלט עבור קלט שונה.

```
import math
A=input("Enter r for radius, d for diameter, l for length, s for square: ")
val=float(input("Enter the value: "))
match A:
    case 'r':
        print(" diameter = "+str(val/2))
        print(" length = " + str(val*2*math.pi))
```

```

print(" square = " + str(val**2*math.pi))
case 'd':
    print(" radius = " + str(val * 2))
    print(" length = " + str(4*val * math.pi))
    print(" square = " + str((val*2) ** 2 * math.pi))
case 'l':
    print(" radius = " + str(val / 2/math.pi))
    print(" diameter = " + str(val /4/ math.pi))
    print(" square = " + str(val ** 2 /4/ math.pi))
case 's':
    print(" radius = " + str(math.sqrt(val/math.pi)))
    print(" diameter = " + str(math.sqrt(val/math.pi)/2))
    print(" length = " + str(2*math.pi*math.sqrt(val/math.pi)))
case _:
    print("No Such operator")

```

תרגיל מס' 44.

תפענח את הקוד הבא והצג מספר דוגמאות של פלט עבור קלט שונה.

```

A=int(input("Enter the first number: "))
B=int(input("Enter the second number: "))
i=input("Enter the sign: ")
match i:
    case '+':
        print("A+B="+str(A+B))
    case '-':
        print("A-B="+str(A-B))
    case '*':
        print("A*B="+str(A*B))
    case '/':
        if(B!=0):
            print("A/B=" + str(A / B))
        else:
            print("Can't divide by zero ")
    case _:
        print("No Such operator")

```

תרגיל מס' 45.

תפענח את הקוד הבא והצג מספר דוגמאות של פלט עבור קלט שונה.

```

A=int(input("Enter the length: "))
B=input("Enter initial units - c (cm), d (dm) m (m) k (km): ")
C=input("Enter destination units - c (cm), d (dm) m (m) k (km): ")

```



```

match B:
  case 'c'|'C':
    match C:
      case 'c'|'C':
        print("The value in cm is "+ str(A))
      case 'd'|'D':
        print("The value in dm is " + str(A/10))
      case 'm'|'M':
        print("The value in m is " + str(A/100))
      case 'k'|'K':
        print("The value in km is " + str(A / 100000))
      case _:
        print("No Such Unit")
  case 'd' | 'D':
    match C:
      case 'c' | 'C':
        print("The value in cm is " + str(A*10))
      case 'd' | 'D':
        print("The value in dm is " + str(A))
      case 'm' | 'M':
        print("The value in m is " + str(A / 10))
      case 'k' | 'K':
        print("The value in km is " + str(A / 10000))
      case _:
        print("No Such Unit")
  case 'm' | 'M':
    match C:
      case 'c' | 'C':
        print("The value in cm is " + str(A * 100))
      case 'd' | 'D':
        print("The value in dm is " + str(A*10))
      case 'm' | 'M':
        print("The value in m is " + str(A))
      case 'k' | 'K':
        print("The value in km is " + str(A / 1000))
      case _:
        print("No Such Unit")
  case 'k' | 'K':
    match C:
      case 'c' | 'C':
        print("The value in cm is " + str(A * 100000))
      case 'd' | 'D':
        print("The value in dm is " + str(A * 10000))
      case 'm' | 'M':

```

```

    print("The value in m is " + str(A * 1000))
case 'k' | 'K':
    print("The value in km is " + str(A))
case _:
    print("No Such Unit")
case _:
    print("No Such Unit")

```

תרגיל מס' 46.

תפענח את הקוד הבא והצג מספר דוגמאות של פלט עבור קלט שונה.

```

A=int(input("Enter the weight: "))
B=input("Enter initial units - m (mgr), g (gr), k (kgr), t (ton): ")
C=input("Enter destination units - m (mgr), g (gr), k (kgr), t (ton): ")
match B:
    case 'm'|'M':
        match C:
            case 'm'|'M':
                print("The value in milli gram is "+ str(A))
            case 'g'|'G':
                print("The value in gram is " + str(A/1000))
            case 'k'|'K':
                print("The value in kilogram is " + str(A/1000000))
            case 't'|'T':
                print("The value in ton is " + str(A / 1000000000))
            case _:
                print("No Such Unit")
    case 'g' | 'G':
        match C:
            case 'm' | 'M':
                print("The value in milli gram is " + str(A*1000))
            case 'g' | 'G':
                print("The value in gram is " + str(A))
            case 'k' | 'K':
                print("The value in kilogram is " + str(A / 1000))
            case 't' | 'T':
                print("The value in ton is " + str(A / 1000000))
            case _:
                print("No Such Unit")
    case 'k' | 'K':
        match C:
            case 'm' | 'M':
                print("The value in milli gram is " + str(A*1000000))
            case 'g' | 'G':

```

```

    print("The value in gram is " + str(A * 1000))
case 'k' | 'K':
    print("The value in kilogram is " + str(A ))
case 't' | 'T':
    print("The value in ton is " + str(A / 1000))
case _:
    print("No Such Unit")
case 't' | 'T':
    match C:
        case 'm' | 'M':
            print("The value in milli gram is " + str(A*1000000000))
        case 'g' | 'G':
            print("The value in gram is " + str(A * 1000000))
        case 'k' | 'K':
            print("The value in kilogram is " + str(A * 1000))
        case 't' | 'T':
            print("The value in ton is " + str(A ))
        case _:
            print("No Such Unit")
case _:
    print("No Such Unit")

```

תרגיל מס' 47.

תפענח את הקוד הבא והצג מספר דוגמאות של פלט עבור קלט שונה.

```

B=int(input("Enter the date to get tomorrow and yesterday: "))
A=int(input("Enter the month to get tomorrow and yesterday: "))
match A:
    case 1:
        match B:
            case 1:
                print("Tomorrow will " + str(B+1) + " of " + str(A))
                print("Yesterday was " + str(31) + " of " + str(12))
            case 31:
                print("Tomorrow will " + str(1) + " of " + str(A+1))
                print("Yesterday was " + str(B-1) + " of " + str(12))
            case B if B<31:
                print("Tomorrow will " + str(B + 1) + " of " + str(A))
                print("Yesterday was " + str(B - 1) + " of " + str(A))
            case _:
                print("No such date")
    case 2:
        match B:
            case 1:

```

```

    print("Tomorrow will " + str(B + 1) + " of " + str(A))
    print("Yesterday was " + str(31) + " of " + str(A-1))
case 28|29:
    print("Tomorrow will " + str(1) + " of " + str(A + 1))
    print("Yesterday was " + str(B - 1) + " of " + str(A-1))
case B if B<28:
    print("Tomorrow will " + str(B + 1) + " of " + str(A))
    print("Yesterday was " + str(B - 1) + " of " + str(A))
case _:
    print("No such date")
case 3:
    match B:
        case 1:
            print("Tomorrow will " + str(B + 1) + " of " + str(A))
            print("Yesterday was " + str(28) + " of " + str(A-1))
        case 30:
            print("Tomorrow will " + str(1) + " of " + str(A + 1))
            print("Yesterday was " + str(B - 1) + " of " + str(A))
        case B if B<31:
            print("Tomorrow will " + str(B + 1) + " of " + str(A))
            print("Yesterday was " + str(B - 1) + " of " + str(A))
        case _:
            print("No such date")
case 5|7|8|10:
    match B:
        case 1:
            print("Tomorrow will " + str(B + 1) + " of " + str(A))
            print("Yesterday was " + str(30) + " of " + str(A - 1))
        case 31:
            print("Tomorrow will " + str(1) + " of " + str(A + 1))
            print("Yesterday was " + str(B - 1) + " of " + str(A - 1))
        case B if B<31:
            print("Tomorrow will " + str(B + 1) + " of " + str(A))
            print("Yesterday was " + str(B - 1) + " of " + str(A))
        case _:
            print("No such date")
case 4 | 6 | 9 | 11:
    match B:
        case 1:
            print("Tomorrow will " + str(B + 1) + " of " + str(A))
            print("Yesterday was " + str(31) + " of " + str(A - 1))
        case 30:
            print("Tomorrow will " + str(1) + " of " + str(A + 1))
            print("Yesterday was " + str(B - 1) + " of " + str(A - 1))

```

```

case B if B<30:
    print("Tomorrow will " + str(B + 1) + " of " + str(A))
    print("Yesterday was " + str(B - 1) + " of " + str(A))
case _:
    print("No such date")
case 12:
    match B:
        case 1:
            print("Tomorrow will " + str(B + 1) + " of " + str(A))
            print("Yesterday was " + str(30) + " of " + str(A-1))
        case 31:
            print("Tomorrow will " + str(1) + " of " + str(1))
            print("Yesterday was " + str(B - 1) + " of " + str(A))
        case B if B<31:
            print("Tomorrow will " + str(B + 1) + " of " + str(A))
            print("Yesterday was " + str(B - 1) + " of " + str(A))
        case _:
            print("No such date")
case _:
    print("No Such Month")

```

תרגיל מס' 48.

תפענח את הקוד הבא והצג מספר דוגמאות של פלט עבור קלט שונה.

```

A=int(input("Enter the first number: "))
B=int(input("Enter the second number: "))
i=input("Enter the sign: ")
match i:
    case '+':
        print("A+B="+str(A+B))
    case '-':
        print("A-B="+str(A-B))
    case '*':
        print("A*B="+str(A*B))
    case '/':
        match B:
            case 0:
                print("Can't divide by zero ")
            case _:
                print("A/B=" + str(A / B))
    case _:
        print("No Such operator")

```

תפענח את הקוד הבא והצג מספר דוגמאות של פלט עבור קלט שונה.

```
import math
A=input("Enter value convert from r for radius, d for diamter, l for length, s for square: ")
B=input("Enter value convert to r for radius, d for diamter, l for length, s for square: ")
val=float(input("Enter the value: "))
match A:
    case 'r'|'R':
        match B:
            case 'd'|'D':
                print(" diameter = "+str(val/2))
            case 'l'|'L':
                print(" length = " + str(val*2*math.pi))
            case 's'|'S':
                print(" square = " + str(val**2*math.pi))
            case _:
                print("No Such operator")
    case 'd'|'D':
        match B:
            case 'r' | 'R':
                print(" radius = " + str(val * 2))
            case 'l' | 'L':
                print(" length = " + str(4*val * math.pi))
            case 's'|'S':
                print(" square = " + str((val*2) ** 2 * math.pi))
            case _:
                print("No Such operator")
    case 'l':
        match B:
            case 'r' | 'R':
                print(" radius = " + str(val / 2/math.pi))
            case 'd' | 'D':
                print(" diameter = " + str(val /4/ math.pi))
            case 's' | 'S':
                print(" square = " + str(val ** 2 /4/ math.pi))
            case _:
                print("No Such operator")
    case 's':
        match B:
            case 'r' | 'R':
                print(" radius = " + str(math.sqrt(val/math.pi)))
            case 'd' | 'D':
                print(" diameter = " + str(math.sqrt(val/math.pi)/2))
```

```

case 'l' | 'L':
    print(" length = " + str(2*math.pi*math.sqrt(val/math.pi)))
case _:
    print("No Such operator")
case _:
    print("No Such operator")

```

תרגיל מס' 50.

תפענח את הקוד הבא והצג מספר דוגמאות של פלט עבור קלט שונה.

```

import math
A=input("Enter value convert from: a for catetus, h for height, c for hypotenuse, s for square: ")
B=input("Enter value convert to: a for catetus, h for height, c for hypotenuse, s for square: ")
val=float(input("Enter the value: "))
match A:
    case 'a'|'A':
        match B:
            case 'h'|'H':
                print(" height is = "+str(val/math.sqrt(2)))
            case 'c'|'C':
                print(" hypotenuse = " + str(val*math.sqrt(2)))
            case 's'|'S':
                print(" square = " + str(val**2/2))
            case _:
                print("No Such operator")
    case 'h'|'H':
        match B:
            case 'a' | 'A':
                print(" catetus = " + str(val *math.sqrt(2)))
            case 'c' | 'C':
                print(" hypotenuse = " + str(2*val))
            case 's'|'S':
                print(" square = " + str(val**2))
            case _:
                print("No Such operator")
    case 'c'|'C':
        match B:
            case 'a' | 'a':
                print(" catetus = " + str(val / math.sqrt(2)))
            case 'h' | 'H':
                print(" height = " + str(val /2))
            case 's' | 'S':
                print(" square = " + str(val ** 2 /4))
            case _:

```

```

        print("No Such operator")
    case 's'|'S':
        match B:
            case 'a' | 'A':
                print(" catetus = " + str(math.sqrt(val*2)))
            case 'c' | 'C':
                print(" height = " + str(math.sqrt(val)))
            case 'h' | 'H':
                print(" hypotenuse = " + str(math.sqrt(val)*2))
            case _:
                print("No Such operator")
    case _:
        print("No Such operator")

```

תרגיל מס' 51.

תפענח את הקוד הבא והצג מספר דוגמאות של פלט עבור קלט שונה.

```

import random
card=random.randint(0,52)
print(card)
color=card//13
type=card%13
match type:
    case 0:
        print("two",end=' ')
    case 1:
        print("three", end=' ')
    case 2:
        print("four", end=' ')
    case 3:
        print("five", end=' ')
    case 4:
        print("six", end=' ')
    case 5:
        print("seven", end=' ')
    case 6:
        print("eight", end=' ')
    case 7:
        print("nine", end=' ')
    case 8:
        print("ten", end=' ')
    case 9:
        print("Jack", end=' ')
    case 10:

```



```

    print("Queen", end=' ')
case 11:
    print("King", end=' ')
case 12:
    print("Ace", end=' ')
match color:
case 0:
    print("Spades")
case 1:
    print("Clubs")
case 2:
    print("Diamonds")
case 3:
    print("Hearts")

```

תרגיל מס' 52.

תפענח את הקוד הבא והצג מספר דוגמאות של פלט עבור קלט שונה. (נלמד רשימות בעתיד – בינתיים נתייחס כי רשימה אקראית).

```

import sys
import random
List = ['a', 0, 2]
a=random.randint(0,2)
b=List[a]
try:
    print("The entry is",b)
    r = 1/int(b)
    print("The answ is", r)
except:
    print("Oops!", sys.exc_info()[0], "occurred.")

```

תרגיל מס' 53.

תפענח את הקוד הבא והצג מספר דוגמאות של פלט עבור קלט שונה.

```

try:
    num = int(input("Enter the num: "))
    assert num % 2 == 0 except:
        print("The num is odd!")
else:
    try:
        reciprocal = 1/num
        print(reciprocal)
    except:
        print("Divide by zero")

```

תפענח את הקוד הבא והצג מספר דוגמאות של פלט עבור קלט שונה.

```
try:
    a = float(input("Enter the number: "))
    b = float(input("Enter divisor: "))
    c = a / b
    print("Answer: %.2f" % c)
except ValueError:
    print("You can't enter string")
except ZeroDivisionError:
    print("You can't divide by zero")
```

תפענח את הקוד הבא והצג מספר דוגמאות של פלט עבור קלט שונה.

```
try:
    a=input("Enter a")
    b=input("Enter b")
    c=input("Enter c")
    try:
        if (a + b - c) < 0:
            print(a + 5)
        else:
            print("Hi" + "4")
    except (AttributeError, TypeError) as e:
        print("Error:", e)
except:
    print("Error of input")
finally:
    print("try except executed")
```

## הפיכת קוד מצורה אחד לצורה אחרת. תרגיל מס' 1.

תהפוך את הקוד הבא לקוד בשימוש במשפט `if, elif, else`. שים לב שקוד צריך להיות ככל שיותר יעיל ומצומצם.

```
n=int(input("Enter the mark from 1 to 10: "))
match n:
    case 1|2|3:
        print("Troll")
    case 4|5:
        print("Bad mark")
    case 6|7:
        print("Not Bad")
    case 8|9:
        print("Good")
    case 10:
        print("Excellent")
    case _:
        print("No Such mark")
```

## תרגיל מס' 2.

תהפוך את הקוד הבא לקוד בשימוש במשפט `match case`. שים לב שקוד צריך להיות ככל שיותר יעיל ומצומצם.

```
ch=int(input("What You want to know? Mass - 1, Volume - 2 Density - 3"))
if ch==1:
    v=float(input("Enter the volume "))
    ro=float(input("Enter the density "))
    m=v*ro
elif ch==2:
    m = float(input("Enter the mass "))
    ro = float(input("Enter the density "))
    v=m/ro
else:
    v = float(input("Enter the volume "))
    m = float(input("Enter the mass "))
    ro=m/v
print("Mass = " + str(m) + " Denisty = " + str(ro) + " Volume " + str(v) )
```

### תרגיל מס' 3.

תהפוך את הקוד הבא לקוד בשימוש במשפט match case. שים לב שקוד צריך להיות ככל שיותר יעיל ומצומצם.

```
import math
ch=int(input("What figure are You talking about? Rectangle - 1, Circle - 2, Triangle - 3 "))
if ch==1:
    a=float(input("Enter first side "))
    b=float(input("Enter second side "))
    S=a*b
elif ch==2:
    r = float(input("Enter the radius "))
    S=math.pi*r*r
else:
    a = float(input("Enter first side "))
    b = float(input("Enter second side "))
    c = float(input("Enter third side "))
    p=(a+b+c)/2
    S=math.sqrt(p*(p-a)*(p-b)*(p-c))
print("Square is " + str(S))
```

### תרגיל מס' 4.

תהפוך את הקוד הבא לקוד בשימוש במשפט match case. שים לב שקוד צריך להיות ככל שיותר יעיל ומצומצם.

```
x = float(input("Enter x "))
y = float(input("Enter y "))
r = input("Enter operator ")
c=0
if r=='+' :
    c=x+y
elif r=='-' :
    c=x-y
elif r=='*' :
    c=x*y
elif r=='/' :
    c=x/y
else:
    print("Unknown function")
print(c)
```

## תרגיל מס' 5.

תהפוך את הקוד הבא לקוד בשימוש במשפט match case. שים לב שקוד צריך להיות ככל שיותר יעיל ומצומצם.

```
flag = input(" What You want to check? (m, d, v): ")
result = 0
if flag == 'm':
    d = float(input(" Density "))
    v = float(input(" Volume "))
    result = d * v
elif flag == 'd':
    m = float(input(" Mass "))
    v = float(input(" Volume "))
    result = m / v
elif flag == 'v':
    m = float(input(" Mass "))
    d = float(input(" Density "))
    result = m / d
print("%.2f" % result)
```

## תרגיל מס' 6.

תהפוך את הקוד הבא לקוד בשימוש במשפט match case. שים לב שקוד צריך להיות ככל שיותר יעיל ומצומצם.

```
print("Enter x and y:")
x = float(input("x = "))
y = float(input("y = "))
if x == 0 and y == 0:
    print("At the center")
elif x == 0:
    print("Axis 'x'")
elif y == 0:
    print("Axis 'y'")
elif x > 0:
    if y > 0:
        print(" I quarter")
    elif y < 0:
        print("IV quarter")
elif x < 0:
    if y > 0:
        print("II quarter")
    elif y < 0:
        print("III quarter")
```

תרגיל מס' 7.

תהפוך את הקוד הבא לקוד בשימוש במשפט match case. שים לב שקוד צריך להיות ככל שיותר יעיל ומצומצם.

```
temper=int(input("Please enter the temperature "))
tempt=input("Please enter type of temperature F or C ")
if(tempt=="F" or tempt=="f"):
    tempC=float((temper-32)*5/9)
    print(tempC)
elif (tempt=="C" or tempt=="c"):
    tempF=float(temper*9/5+32)
    print(tempF)
else:
    print("Unknown letter")
```

תרגיל מס' 8.

תהפוך את הקוד הבא לקוד בשימוש במשפט match case. שים לב שקוד צריך להיות ככל שיותר יעיל ומצומצם.

```
print("1-rectangle, 2-triangle, 3-circle")
figure = input("Choose one of them: ")
if figure == '1':
    print("Enter sides of rectangle:")
    a = float(input("a = "))
    b = float(input("b = "))
    print("Square is: %.2f" % (a * b))
elif figure == '2':
    print("Enter 3 sides of triangle:")
    a = float(input("a = "))
    b = float(input("b = "))
    c = float(input("c = "))
    p = (a + b + c) / 2
    from math import sqrt
    s = sqrt(p * (p - a) * (p - b) * (p - c))
    print("Square is: %.2f" % s)
elif figure == '3':
    r = float(input("Radius of the circle R = "))
    from math import pi
    print("Square is: %.2f" % (pi * r ** 2))
else:
    print("Error of input")
```

תהפוך את הקוד הבא לקוד בשימוש במשפט match case. שים לב שקוד צריך להיות ככל שיותר יעיל ומצומצם.

```
s = input("Enter the sign (+,-,*,/): ")
if s in ('+', '-', '*', '/'):
    x = float(input("x="))
    y = float(input("y="))
    if s == '+':
        print("%.2f" % (x+y))
    elif s == '-':
        print("%.2f" % (x-y))
    elif s == '*':
        print("%.2f" % (x*y))
    elif s == '/':
        if y != 0:
            print("%.2f" % (x/y))
        else:
            print("You cant divide by zero!")
    else:
        print("Illegal sign")
```

תהפוך את הקוד הבא לקוד בשימוש במשפט match case. שים לב שקוד צריך להיות ככל שיותר יעיל ומצומצם.

```
import math
print("Now we shall solve ax2+bx+c=0")
a=int(input("Please enter a: "))
b=int(input("Please enter b: "))
c=int(input("Please enter c: "))
if a==0:
    if b==0:
        if c!=0:
            print("No roots")
        else:
            print("Infinite roots")
    else:
        x=-c/b
        print("x= " + str(x))
elif b==0:
    if -c/a>0:
        x1= math.sqrt(-c/a)
        x2=-x1
```

```

print(" x1= " + str(x1)+ " x2= " + str(x2))
else:
    print("Complex roots")
elif c==0:
    x1 = 0
    x2 = -b/a
    print(" x1= " + str(x1) + " x2= " + str(x2))
else:
    D=b*b-4*a*c
    if D<0:
        print("Complex roots")
    elif D==0:
        x=-b/(2*a)
        print("x= " + str(x))
    else:
        x1=(-b+math.sqrt(D))/(2*a)
        x2=(-b-math.sqrt(D))/(2*a)
        print(" x1= " + str(x1) + " x2= " + str(x2))

```

תרגיל מס' 11.

תהפוך את הקוד הבא לקוד בשימוש במשפט `match case`. שים לב שקוד צריך להיות ככל שיותר יעיל ומצומצם.

```

import random
A = random.randrange(-3,3)
B = random.randrange(-3,3)
print("Number A:", A)
print("Number B:", B)
if A != B:
    A = B = max(A,B)
else:
    A = B = 0
print("Number A:", A)
print("Number B:", B)

```

תרגיל מס' 12.

תהפוך את הקוד הבא לקוד בשימוש במשפט `if, elif, else`. שים לב שקוד צריך להיות ככל שיותר יעיל ומצומצם.

```

import math
A=input("Enter value convert from: a for catetus, h for height, c for hypotenuse, s for square: ")
val=float(input("Enter the value: "))
match A:

```



```

case 'a'|'A':
    print(" height is = "+str(val/math.sqrt(2)))
    print(" hypotenuse = " + str(val*math.sqrt(2)))
    print(" square = " + str(val**2/2))
case 'h'|'H':
    print(" catetus = " + str(val *math.sqrt(2)))
    print(" hypotenuse = " + str(2*val))
    print(" square = " + str(val**2))
case 'c'|'C':
    print(" catetus = " + str(val / math.sqrt(2)))
    print(" height = " + str(val /2))
    print(" square = " + str(val ** 2 /4))
case 's'|'S':
    print(" catetus = " + str(math.sqrt(val*2)))
    print(" height = " + str(math.sqrt(val)))
    print(" hypotenuse = " + str(math.sqrt(val)*2))
case _:
    print("No Such operator")

```

תרגיל מס' 13.

תהפוך את הקוד הבא לקוד בשימוש במשפט if, elif, else. שים לב שקוד צריך להיות ככל שיותר יעיל ומצומצם.

```

A=int(input("Enter the month of birthday: "))
B=int(input("Enter the date of birthday: "))
match A:
    case 1:
        if 0<B<=19:
            print("You are Capricornus")
        elif B<=31:
            print("You are Aquarius")
        else:
            print("No Such Day")
    case 2:
        if 0 < B <= 18:
            print("You are Aquarius")
        elif B <= 28:
            print("You are Pisces")
        else:
            print("No Such Day")
    case 3:
        if 0 < B <= 18:
            print("You are Pisces")
        elif B <= 31:

```

```

    print("You are Aries")
else:
    print("No Such Day")
case 4:
    if 0 < B <= 19:
        print("You are Aries")
    elif B <= 30:
        print("You are Taurus")
    else:
        print("No Such Day")
case 5:
    if 0 < B <= 20:
        print("You are Taurus")
    elif B <= 31:
        print("You are Gemini")
    else:
        print("No Such Day")
case 6:
    if 0 < B <= 21:
        print("You are Gemini")
    elif B <= 30:
        print("You are Cancer")
    else:
        print("No Such Day")
case 7:
    if 0 < B <= 22:
        print("You are Cancer")
    elif B <= 31:
        print("You are Leo")
    else:
        print("No Such Day")
case 8:
    if 0 < B <= 22:
        print("You are Leo")
    elif B <= 31:
        print("You are Virgo")
    else:
        print("No Such Day")
case 9:
    if 0 < B <= 22:
        print("You are Virgo")
    elif B <= 30:
        print("You are Libra")
    else:

```

```

    print("No Such Day")
case 10:
    if 0 < B <= 23:
        print("You are Libra")
    elif B <= 31:
        print("You are Scorpius")
    else:
        print("No Such Day")
case 11:
    if 0 < B <= 21:
        print("You are Scorpius")
    elif B <= 30:
        print("You are Sagittarius")
    else:
        print("No Such Day")
case 12:
    if 0 < B <= 21:
        print("You are Sagittarius")
    elif B <= 28:
        print("You are Capricornus")
    else:
        print("No Such Day")
case _:
    print("No Such Month")

```

תרגיל מס' 14.

תהפוך את הקוד הבא לקוד בשימוש במשפט if, elif, else. שים לב שקוד צריך להיות ככל שיותר יעיל ומצומצם.

```

A=int(input("Enter the year of birthday after 1920: "))
A-=1920
A%=12
match A:
    case 0:
        print("Monkey")
    case 1:
        print("Rooster")
    case 2:
        print("Dog")
    case 3:
        print("Pig")
    case 4:
        print("Rat")
    case 5:

```

```

print("Ox")
case 6:
    print("Tiger")
case 7:
    print("Rabbit")
case 8:
    print("Dragon")
case 9:
    print("Snake")
case 10:
    print("Horse")
case 11:
    print("Goat")

```

תרגיל מס' 15.

תהפוך את הקוד הבא לקוד בשימוש במשפט `if, elif, else`. שים לב שקוד צריך להיות ככל שיותר יעיל ומצומצם.

```

n=int(input("Enter the num: "))
match n:
    case 1:
        print("Sunday")
    case 2:
        print("Monday")
    case 3:
        print("Tuesday")
    case 4:
        print("Wednesday")
    case 5:
        print("Thursday")
    case 6:
        print("Friday")
    case 7:
        print("Saturday")
    case _:
        print("No Such Day")

```

תרגיל מס' 16.

תהפוך את הקוד הבא לקוד בשימוש במשפט `if, elif, else`. שים לב שקוד צריך להיות ככל שיותר יעיל ומצומצם.

```

n=int(input("Enter the mark from 1 to 10: "))
match n:

```

```

case 1|2|3:
    print("Troll")
case 4|5:
    print("Bad mark")
case 6|7:
    print("Not Bad")
case 8|9:
    print("Good")
case 10:
    print("Excellent")
case _:
    print("No Such mark")

```

תרגיל מס' 17.

תהפוך את הקוד הבא לקוד בשימוש במשפט if, elif, else. שים לב שקוד צריך להיות ככל שיותר יעיל ומצומצם.

```

n=int(input("Enter the month from 1 to 12: "))
match n:
    case 12|1|2:
        print("Winter")
    case 3|4|5:
        print("Spring")
    case 6|7|8:
        print("Summer")
    case 9|10|11:
        print("Autumn")
    case _:
        print("No Such month")

```

תרגיל מס' 18.

תהפוך את הקוד הבא לקוד בשימוש במשפט if, elif, else. שים לב שקוד צריך להיות ככל שיותר יעיל ומצומצם.

```

n=int(input("Enter the month from 1 to 12: "))
match n:
    case 1|3|5|7|8|10|12:
        print("31 days")
    case 4|6|9|11:
        print("30 days")
    case 2:
        print("28-29 Days")
    case _:
        print("No Such month")

```

תהפוך את הקוד הבא לקוד בשימוש במשפט if, elif, else. שים לב שקוד צריך להיות ככל שיותר יעיל ומצומצם.

```
import math
A=input("Enter r for radius, d for diameter, l for length, s for square: ")
val=float(input("Enter the value: "))
match A:
    case 'r':
        print(" diameter = "+str(val/2))
        print(" length = " + str(val*2*math.pi))
        print(" square = " + str(val**2*math.pi))
    case 'd':
        print(" radius = " + str(val * 2))
        print(" length = " + str(4*val * math.pi))
        print(" square = " + str((val*2) ** 2 * math.pi))
    case 'l':
        print(" radius = " + str(val / 2/math.pi))
        print(" diameter = " + str(val /4/ math.pi))
        print(" square = " + str(val ** 2 /4/ math.pi))
    case 's':
        print(" radius = " + str(math.sqrt(val/math.pi)))
        print(" diameter = " + str(math.sqrt(val/math.pi)/2))
        print(" length = " + str(2*math.pi*math.sqrt(val/math.pi)))
    case _:
        print("No Such operator")
```

תהפוך את הקוד הבא לקוד בשימוש במשפט if, elif, else. שים לב שקוד צריך להיות ככל שיותר יעיל ומצומצם.

```
A=int(input("Enter the first number: "))
B=int(input("Enter the second number: "))
i=input("Enter the sign: ")
match i:
    case '+':
        print("A+B="+str(A+B))
    case '-':
        print("A-B="+str(A-B))
    case '*':
        print("A*B="+str(A*B))
    case '/':
        if(B!=0):
            print("A/B=" + str(A / B))
```

```

else:
    print("Can't divide by zero ")
case _:
    print("No Such operator")

```

תרגיל מס' 21.

תהפוך את הקוד הבא לקוד בשימוש במשפט if, elif, else. שים לב שקוד צריך להיות ככל שיותר יעיל ומצומצם.

```

A=int(input("Enter the length: "))
B=input("Enter initial units - c (cm), d (dm) m (m) k (km): ")
C=input("Enter destination units - c (cm), d (dm) m (m) k (km): ")
match B:
    case 'c'|'C':
        match C:
            case 'c'|'C':
                print("The value in cm is " + str(A))
            case 'd'|'D':
                print("The value in dm is " + str(A/10))
            case 'm'|'M':
                print("The value in m is " + str(A/100))
            case 'k'|'K':
                print("The value in km is " + str(A / 100000))
            case _:
                print("No Such Unit")
    case 'd' | 'D':
        match C:
            case 'c' | 'C':
                print("The value in cm is " + str(A*10))
            case 'd' | 'D':
                print("The value in dm is " + str(A))
            case 'm' | 'M':
                print("The value in m is " + str(A / 10))
            case 'k' | 'K':
                print("The value in km is " + str(A / 10000))
            case _:
                print("No Such Unit")
    case 'm' | 'M':
        match C:
            case 'c' | 'C':
                print("The value in cm is " + str(A * 100))
            case 'd' | 'D':
                print("The value in dm is " + str(A*10))
            case 'm' | 'M':

```

```

    print("The value in m is " + str(A))
case 'k' | 'K':
    print("The value in km is " + str(A / 1000))
case _:
    print("No Such Unit")
case 'k' | 'K':
    match C:
        case 'c' | 'C':
            print("The value in cm is " + str(A * 100000))
        case 'd' | 'D':
            print("The value in dm is " + str(A * 10000))
        case 'm' | 'M':
            print("The value in m is " + str(A * 1000))
        case 'k' | 'K':
            print("The value in km is " + str(A))
        case _:
            print("No Such Unit")
case _:
    print("No Such Unit")

```

תרגיל מס' 22.

תהפוך את הקוד הבא לקוד בשימוש במשפט if, elif, else. שים לב שקוד צריך להיות ככל שיותר יעיל ומצומצם.

```

A=int(input("Enter the weight: "))
B=input("Enter initial units - m (mgr), g (gr), k (kgr), t (ton): ")
C=input("Enter destination units - m (mgr), g (gr), k (kgr), t (ton): ")
match B:
    case 'm'|'M':
        match C:
            case 'm'|'M':
                print("The value in milli gram is "+ str(A))
            case 'g'|'G':
                print("The value in gram is " + str(A/1000))
            case 'k'|'K':
                print("The value in kilogram is " + str(A/1000000))
            case 't'|'T':
                print("The value in ton is " + str(A / 1000000000))
            case _:
                print("No Such Unit")
    case 'g' | 'G':
        match C:
            case 'm' | 'M':
                print("The value in milli gram is " + str(A*1000))

```



```

case 'g' | 'G':
    print("The value in gram is " + str(A))
case 'k' | 'K':
    print("The value in kilogram is " + str(A / 1000))
case 't' | 'T':
    print("The value in ton is " + str(A / 1000000))
case _:
    print("No Such Unit")
case 'k' | 'K':
    match C:
        case 'm' | 'M':
            print("The value in milli gram is " + str(A*1000000))
        case 'g' | 'G':
            print("The value in gram is " + str(A * 1000))
        case 'k' | 'K':
            print("The value in kilogram is " + str(A ))
        case 't' | 'T':
            print("The value in ton is " + str(A / 1000))
        case _:
            print("No Such Unit")
case 't' | 'T':
    match C:
        case 'm' | 'M':
            print("The value in milli gram is " + str(A*1000000000))
        case 'g' | 'G':
            print("The value in gram is " + str(A * 1000000))
        case 'k' | 'K':
            print("The value in kilogram is " + str(A * 1000))
        case 't' | 'T':
            print("The value in ton is " + str(A ))
        case _:
            print("No Such Unit")
case _:
    print("No Such Unit")

```

תרגיל מס' 23.

תהפוך את הקוד הבא לקוד בשימוש במשפט if, elif, else. שים לב שקוד צריך להיות ככל שיותר יעיל ומצומצם.

```

B=int(input("Enter the date to get tomorrow and yesterday: "))
A=int(input("Enter the month to get tomorrow and yesterday: "))
match A:
    case 1:
        match B:

```

```

case 1:
    print("Tomorrow will " + str(B+1) + " of " + str(A))
    print("Yesterday was " + str(31) + " of " + str(12))
case 31:
    print("Tomorrow will " + str(1) + " of " + str(A+1))
    print("Yesterday was " + str(B-1) + " of " + str(12))
case B if B<31:
    print("Tomorrow will " + str(B + 1) + " of " + str(A))
    print("Yesterday was " + str(B - 1) + " of " + str(A))
case _:
    print("No such date")
case 2:
    match B:
        case 1:
            print("Tomorrow will " + str(B + 1) + " of " + str(A))
            print("Yesterday was " + str(31) + " of " + str(A-1))
        case 28|29:
            print("Tomorrow will " + str(1) + " of " + str(A + 1))
            print("Yesterday was " + str(B - 1) + " of " + str(A-1))
        case B if B<28:
            print("Tomorrow will " + str(B + 1) + " of " + str(A))
            print("Yesterday was " + str(B - 1) + " of " + str(A))
        case _:
            print("No such date")
case 3:
    match B:
        case 1:
            print("Tomorrow will " + str(B + 1) + " of " + str(A))
            print("Yesterday was " + str(28) + " of " + str(A-1))
        case 30:
            print("Tomorrow will " + str(1) + " of " + str(A + 1))
            print("Yesterday was " + str(B - 1) + " of " + str(A))
        case B if B<31:
            print("Tomorrow will " + str(B + 1) + " of " + str(A))
            print("Yesterday was " + str(B - 1) + " of " + str(A))
        case _:
            print("No such date")
case 5|7|8|10:
    match B:
        case 1:
            print("Tomorrow will " + str(B + 1) + " of " + str(A))
            print("Yesterday was " + str(30) + " of " + str(A - 1))
        case 31:
            print("Tomorrow will " + str(1) + " of " + str(A + 1))

```

```

    print("Yesterday was " + str(B - 1) + " of " + str(A - 1))
case B if B<31:
    print("Tomorrow will " + str(B + 1) + " of " + str(A))
    print("Yesterday was " + str(B - 1) + " of " + str(A))
case _:
    print("No such date")
case 4 | 6 | 9 | 11:
    match B:
        case 1:
            print("Tomorrow will " + str(B + 1) + " of " + str(A))
            print("Yesterday was " + str(31) + " of " + str(A - 1))
        case 30:
            print("Tomorrow will " + str(1) + " of " + str(A + 1))
            print("Yesterday was " + str(B - 1) + " of " + str(A - 1))
        case B if B<30:
            print("Tomorrow will " + str(B + 1) + " of " + str(A))
            print("Yesterday was " + str(B - 1) + " of " + str(A))
        case _:
            print("No such date")
case 12:
    match B:
        case 1:
            print("Tomorrow will " + str(B + 1) + " of " + str(A))
            print("Yesterday was " + str(30) + " of " + str(A-1))
        case 31:
            print("Tomorrow will " + str(1) + " of " + str(1))
            print("Yesterday was " + str(B - 1) + " of " + str(A))
        case B if B<31:
            print("Tomorrow will " + str(B + 1) + " of " + str(A))
            print("Yesterday was " + str(B - 1) + " of " + str(A))
        case _:
            print("No such date")
case _:
    print("No Such Month")

```

תרגיל מס' 24.

תהפוך את הקוד הבא לקוד בשימוש במשפט if, elif, else. שים לב שקוד צריך להיות ככל שיותר יעיל ומצומצם.

```

A=int(input("Enter the first number: "))
B=int(input("Enter the second number: "))
i=input("Enter the sign: ")
match i:
    case '+':

```

```

    print("A+B="+str(A+B))
case '-':
    print("A-B="+str(A-B))
case '*':
    print("A*B="+str(A*B))
case '/':
    match B:
        case 0:
            print("Can't divide by zero ")
        case _:
            print("A/B=" + str(A / B))
case _:
    print("No Such operator")

```

תרגיל מס' 25.

תהפוך את הקוד הבא לקוד בשימוש במשפט `if, elif, else`. שים לב שקוד צריך להיות ככל שיותר יעיל ומצומצם.

```

import math
A=input("Enter value convert from r for radius, d for diamter, l for length, s for square: ")
B=input("Enter value convert to r for radius, d for diamter, l for length, s for square: ")
val=float(input("Enter the value: "))
match A:
    case 'r'|'R':
        match B:
            case 'd'|'D':
                print(" diameter = "+str(val/2))
            case 'l'|'L':
                print(" length = " + str(val*2*math.pi))
            case 's'|'S':
                print(" square = " + str(val**2*math.pi))
            case _:
                print("No Such operator")
    case 'd'|'D':
        match B:
            case 'r' | 'R':
                print(" radius = " + str(val * 2))
            case 'l' | 'L':
                print(" length = " + str(4*val * math.pi))
            case 's'|'S':
                print(" square = " + str((val*2) ** 2 * math.pi))
            case _:
                print("No Such operator")
    case 'l':

```

```

match B:
    case 'r' | 'R':
        print(" radius = " + str(val / 2/math.pi))
    case 'd' | 'D':
        print(" diameter = " + str(val /4/ math.pi))
    case 's' | 'S':
        print(" square = " + str(val ** 2 /4/ math.pi))
    case _:
        print("No Such operator")
case 's':
    match B:
        case 'r' | 'R':
            print(" radius = " + str(math.sqrt(val/math.pi)))
        case 'd' | 'D':
            print(" diameter = " + str(math.sqrt(val/math.pi)/2))
        case 'l' | 'L':
            print(" length = " + str(2*math.pi*math.sqrt(val/math.pi)))
        case _:
            print("No Such operator")
case _:
    print("No Such operator")

```

תרגיל מס' 26.

תהפוך את הקוד הבא לקוד בשימוש במשפט if, elif, else. שים לב שקוד צריך להיות ככל שיותר יעיל ומצומצם.

```

import math
A=input("Enter value convert from: a for catetus, h for height, c for hypotenuse, s for square: ")
B=input("Enter value convert to: a for catetus, h for height, c for hypotenuse, s for square: ")
val=float(input("Enter the value: "))
match A:
    case 'a'|'A':
        match B:
            case 'h'|'H':
                print(" height is = "+str(val/math.sqrt(2)))
            case 'c'|'C':
                print(" hypotenuse = " + str(val*math.sqrt(2)))
            case 's'|'S':
                print(" square = " + str(val**2/2))
            case _:
                print("No Such operator")
    case 'h'|'H':
        match B:
            case 'a' | 'A':

```

```

    print(" catetus = " + str(val *math.sqrt(2)))
case 'c' | 'C':
    print(" hypotenuse = " + str(2*val))
case 's'|'S':
    print(" square = " + str(val**2))
case _:
    print("No Such operator")
case 'c'|'C':
    match B:
        case 'a' | 'a':
            print(" catetus = " + str(val / math.sqrt(2)))
        case 'h' | 'H':
            print(" height = " + str(val /2))
        case 's' | 'S':
            print(" square = " + str(val ** 2 /4))
        case _:
            print("No Such operator")
case 's'|'S':
    match B:
        case 'a' | 'A':
            print(" catetus = " + str(math.sqrt(val*2)))
        case 'c' | 'C':
            print(" height = " + str(math.sqrt(val)))
        case 'h' | 'H':
            print(" hypotenuse = " + str(math.sqrt(val)*2))
        case _:
            print("No Such operator")
case _:
    print("No Such operator")

```

תרגיל מס' 27.

תהפוך את הקוד הבא לקוד בשימוש במשפט if, elif, else. שים לב שקוד צריך להיות ככל שיותר יעיל ומצומצם.

```

import random
card=random.randint(0,52)
print(card)
color=card//13
type=card%13
match type:
    case 0:
        print("two",end=' ')
    case 1:
        print("three", end=' ')

```

```

case 2:
    print("four", end=' ')
case 3:
    print("five", end=' ')
case 4:
    print("six", end=' ')
case 5:
    print("seven", end=' ')
case 6:
    print("eight", end=' ')
case 7:
    print("nine", end=' ')
case 8:
    print("ten", end=' ')
case 9:
    print("Jack", end=' ')
case 10:
    print("Queen", end=' ')
case 11:
    print("King", end=' ')
case 12:
    print("Ace", end=' ')
match color:
case 0:
    print("Spades")
case 1:
    print("Clubs")
case 2:
    print("Diamonds")
case 3:
    print("Hearts")

```

תרגיל מס' 28.

תהפוך את הקוד הבא לקוד בשימוש אופרטור שורה if else.

```

x=int(input("Enter number to check :"))
if x > 5:
    print("greater")
else:
    print("smaller")

```

תרגיל מס' 29.

תהפוך את הקוד הבא לקוד בשימוש במשפט if, else. תוסיף בהתחלה משפט לקליטת ערך של משתנה לבדיקה.

```
print("Go out for a walk" if nice_weather else "watch a movie at home")
```

תרגיל מס' 30.

תהפוך את הקוד הבא לקוד בשימוש במשפט if, else.

```
num=int(input("Enter number to check :"))  
print("Even" if num % 2 == 0 else "Odd")
```

תרגיל מס' 31.

תהפוך את הקוד הבא לקוד בשימוש במשפט if, else.

```
from random import random  
a,b=round(random()*100),round(random()*100)  
print("a= ",a,"b= ",b)  
res="a" if a>b else "b"  
print("the maximum is", res)
```

תרגיל מס' 32.

תהפוך את הקוד הבא לקוד בשימוש במשפט if, else.

```
x, y = int(input("x= ")), int(input("y= "))  
print("x= ",x," y= ",y)  
small = x if x < y else y  
print(small)
```

תרגיל מס' 33.

תהפוך את הקוד הבא לקוד בשימוש אופרטור שורה if else.

```
number = int(input("number = "))  
if number > 2:  
    print("Number is bigger than 2.")  
elif number < 2:  
    print("Number is smaller than 2.")  
else:  
    print("Number is 2.")
```



תרגיל מס' 34.

תהפוך את הקוד הבא לקוד בשימוש במשפט .if, elif, else.

```
import random
a, b, c = random.randrange(1,100), random.randrange(1,100), random.randrange(1,100)
print("a= ", a, "b= ", b, "c= ", c)
max = a if a > b and a>c else b if b>c else c
print(max)
```

תרגיל מס' 35.

תהפוך את הקוד הבא לקוד בשימוש אופרטור שורה .if else.

```
if i>100:
    x=2
elif i<100:
    x=1
else:
    x=0
```

תרגיל מס' 36.

תהפוך את הקוד הבא לקוד בשימוש במשפט .if, elif, else.

```
import random
i = random.randrange(1,200)
x=2 if i>100 else 1 if i<20 else 0
print(i,x)
```

## תרגול בנושא כתיבת קוד.

### תרגיל מס' 1.

נתונה פונקציה הבאה:

$$y = x - 0.5 \text{ כאשר } x > 0$$

$$y = 0 \text{ כאשר } x = 0$$

$$y = |x| \text{ כאשר } x < 0$$

כתוב תוכנית שמקבלת מהמשתמש ערך של  $x$  ומחזירה ערך של  $y$ .

### תרגיל מס' 2.

כתוב תוכנית שמחשבת משקל או צפיפות או נפח לפי שני נתונים שנקלטים מהמשתמש לפי נוסחה:  $m = V * \rho$ . לדוגמה משתמש רוצה נפח תוכנה מבקשת ממנו לתת משקל וצפיפות.

### תרגיל מס' 3.

משתמש בוחר שטח של איזה גוף הוא רוצה לחשב - משולש, מלבן או מעגל. ניתן לחשב שטח

משולש לפי נוסחה  $S = \sqrt{p(p-a)(p-b)(p-c)}$ , שטח של מלבן  $S = ab$ , שטח של מעגל  $S = \pi r^2$ .

### תרגיל מס' 4.

בדיקה האם משולש מתקיים.

במשולש, הסכום של שני הצלעות חייב להיות גדול מהשלישית. אחרת, שני הצלעות פשוט "ישכבו" על השלישי והמשולש לא יתקבל כמצולע שבור סגור. המשתמש מכניס אורכים של שלוש צלעות על התוכנית לקבוע אם משולש יכול להתקיים עבור צלעות אלה. משמעות הדבר היא שצריך להשוות את הסכומים של כל זוגות הצלעות עם הצלע השלישית שנותרה. על מנת שמשולש יתקיים, סכום שתי צלעות חייב להיות תמיד גדול מהצלע השלישית.

מכיוון שיש רק שלוש צלעות, ניתן להכין שלוש אפשרויות להוספת שני צדדים:  $a + b$ ,  $b + c$ ,  $a + c$ . הסכום הראשון מושווה לשאר הצדדים  $c$ , השנייה עם  $a$  והשלישית עם  $b$ . אם לפחות במקרה אחד הסכום מתגלה כשווה לצלע שלישית, אזי ניתן להסיק כי המשולש אינו יכול להתקיים.

### תרגיל מס' 5.

האם נקודה שייכת למעגל?

קבע אם הנקודה עם הקואורדינטות  $(x; y)$  שייכת למעגל הרדיוס  $R$  שבמרכזו נמצא בראשית הצירים. המשתמש מכניס קואורדינטות של הנקודה ורדיוס המעגל.

אם בחרתם נקודה במישור הקואורדינטות, תוכלו לראות שהקואורדינטות שלה בציר  $x$  ו- $y$  הן יתר של משולש ישר זווית. והיתר של המשולש ישר זווית הזה רק מראה את המרחק מהמקור לנקודה. לפיכך, אם אורך היתר הוא פחות מרדיוס המעגל, אז הנקודה תהיה שייכת למעגל; אחרת זה יהיה מעבר.

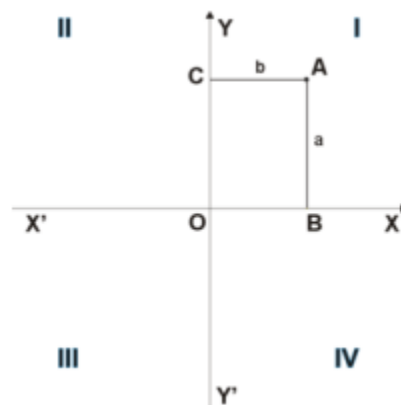
אורך היתר מחושב על ידי משפט פיתגורס: ריבוע היתר שווה לסכום ריבועי הניצבים. מכאן שיתר שווה לשורש הריבועי של סכום ריבועי הניצבים.

תרגיל מס' 6.

כתוב תוכנית שמקבלת שני מספרים עם נקודה עשרונית ופעולה אריתמטית:  $*/-+$ . התוכנה מחזירה תוצאה בהתאם לפעולה מתמטית.

תרגיל מס' 7.

כתוב תוכנית שמקבלת קואורדינטות של נקודה לפי ציר X ו-Y. תוכנה מדפיסה באיזה רבע נמצאת הנקודה.



תרגיל מס' 8.

כתוב תוכנית שמקבלת שנה ומדפיסה האם שנה היא מעוברת או לא.

תרגיל מס' 9.

משתמש מכניס טמפרטורה במעלות צלסיוס או במעלות פרנהייט. התוכנית מחזירה טמפרטורה לפי סקאלה אחרת של טמפרטורה.  $F=C*9/5+32$ ,  $C=(F-32)*5/9$ .

תרגיל מס' 10.

כתוב תוכנית שפותרת משוואה ריבועית לפי  $a, b, c$  שמכניס משתמש. נניח שמקדמים שונים מ-0.

תרגיל מס' 11.

כתוב תוכנית שפותרת משוואה ריבועית לפי  $a, b, c$  שמכניס משתמש. מקדמים יכולים להיות גם שווים ל-0.

תרגיל מס' 12.

כתוב תוכנית לחישוב חשבון חשמל לפי צריכת חשמל, כאשר מחיר מול צריכה נתונה בטבלה הבאה:

מחיר אגורות על kW/h	צריכה kW/h
10	1 עד 100
20	101 עד 200
30	201 עד 300
40	מעל 300

תרגיל מס' 13.

כתוב תוכנית הקולטת מהמשתמש 2 מספרים ומחזירה תוצאת החישוב של  $\log(a/b)$  תוך שימוש במנגנון `try, except`.

תרגיל מס' 14.

קולטים מספר (מיקום) של האות באלף-בית האנגלי מהמקלדת. הצגת התו המבוקש. יש לטפל בכל סוגי שגיאות: קלט לא תקין (A 3.11 -8 וכו'), קלט לא הגיוני - 54.

תרגיל מס' 15.

מהמקלדת מקבלים גבולות לטווח מספרי. הפעל גנרטור של מספר אקראי שלם בתוך התחום והצג אותו. (בדוק מקרה שמספר ראשון שנותנים גדול ממספר שני).

תרגיל מס' 16.

עובד מקבל בונוס של 5% לאחר 5 שנים של עבודה בחברה, לאחר 10 שנים - תוספת של 12.5%. לאחר 15 שנה תוספת של 500 ₪ בנוסף ל-12.5% ולאחר 20 שנה הולך לפנסיה. כתוב תוכנית הקולטת שכר בסיסי, וותק של עובד ומחזירה את הסכום של המשכורת שעובד צריך לקבל. בדוק תקינות הקלט.

תרגיל מס' 17.

כתוב תוכנית הקולטת מהמשתמש 4 צלעות של מרובע ו-4 זוויות ובודקת האם מרובע קיים ומה הוא סוג של מרובע. (מלבן, ריבוע, מקבילית, טרפז, מעוין, דלטון).

תרגיל מס' 18.

כתוב תוכנית הקולטת 3 משתנים מהמשתמש ומחפשת מספר הגדול ביניהם ומספר הקטן ביניהם.

תרגיל מס' 19.

כתוב תוכנית שמחשבת מחיר סופי שלקוח צריך לשלם בהתאם לסכום העסקה וכמות תשלומים:

כל 100 ₪ נוהצג הנחה של 1%. ניתן לשלם עד 10 תשלומים ללא ריבית מעל 1000 ₪, 5 תשלומים מעל 500 ₪. במידה ומשלמים ביותר מ-10 תשלומים מוסיפים 1% לסכום העסקה. יש לקלוט סכום העסקה וכמות תשלומים ולהציג סכום שכל חודש צרכן צריך לשלם.

תרגיל מס' 20.

כתוב תוכנית הקולטת ציון של המשתמש ואחוז נוכחות של הסטודנט. ציון סופי נחשב בתור ציון ועוד בונוס של נוכחות – עד 60 אחוז נוכחות אין בונוס, בין 60 ל-80 אחוז נוכחות תוספת של 5 נקודות, מעל 80 אחוז תוספת של 10 נקודות. שימו לב שציון סופי לא יכול לעלות מעל 100.

תוכנה מחזירה תשובה לאיזה קבוצת ידע הוא שייך:

מעל 80 – A, בין 60 לבין 80 – B, בין 50 ל-60 – C, בין 45 ל-50 – D, בין 25 ל-45 – E, מתחת ל-25 – E.

תרגיל מס' 21.

כתוב תוכנית הקולטת מהמשתמש מספר תלת ספרתי (יש לבדוק תקינות של קלט) ובודקת האם מספר מתחלק ב-3, 6 או 9 ללא שארית. (אם סכום ספרות מתחלק ב-3 - מספר מתחלק ב-3. אם סכום ספרות מתחלק ב-3 ומספר הוא מספר זוגי - מספר מתחלק ב-6, אם סכום ספרות מתחלק ב-9 - מספר מתחלק ב-9).

תרגיל מס' 22.

כתוב תוכנית הקולטת מהמשתמש שלושת הצדדים של המשולש ובמקרה שמשולש חוקי תוכנית מציגה האם זה משולש שווי צלעות, שווי שוקיים או רגיל?

תרגיל מס' 23.

כתוב תוכנית הקולטת מהמשתמש שלושת הצדדים של המשולש ובמקרה שמשולש חוקי תוכנית מציגה האם זה משולש ישר זווית, כהה זווית או רגיל? (יש למצוא קודם מהו הצד הגדול ביותר).

תרגיל מס' 24.

כתוב תוכנית הקולטת מהמשתמש שלושת הזוויות של המשולש ובמקרה שמשולש חוקי תוכנית מציגה האם זה משולש שווי צלעות, שווי שוקיים, ישר זווית, כהה זווית או רגיל?

תרגיל מס' 25.

כתוב תוכנית הקולטת מהמשתמש גובה של אדם, מין (F או M או f או m), גיל ומשקל. טבלת המשקלים לגיל 40 נתונה כאן. במידה ואדם מעל גיל 40 יש להוסיף 1% משקל לגבול עליון על כל 5 שנים במידה ואדם מתחת לגיל 20 טבלה הזאת לא רלוונטית.



משקל מומלץ לגבר	משקל מומלץ לאישה	גובה
60.5-65	55-61.4	1.60
62.3-67.3	57.7-64.1	1.65
64.5-70	60.5-66.8	1.70
67.3-72.7	63.2-69.5	1.75
70-75	66-72	1.80
72-77	67-74	1.83

תרגיל מס' 26.

נתונה טבלה של כמות הצעדים ביום הרצויה בהתאם לגיל ומין. כתוב תוכנית שמקבלת מהמשתמש גיל, מין וכמות הצעדים ומדפיסים הודעה מהו אחוז ביצוע משימה יומית אדם עשה. יש לבדוק תקינות של קלט.

Age	Sex	Steps/day
$\geq 18$ and $< 30$	M	800
	F	750
$\geq 30$ and $\leq 40$	M	700
	F	650

תרגיל מס' 27.

כתוב תוכנית הקולטת מהמשתמש ציונים של מדעים, מתמטיקה, אנגלית ומדעי הרוח. כתוב תוכנית שתמליץ לסטודנט לימודים במסלולים מסוימים בהתאם לתנאים הבאים:

ממוצע מעל 80 אנגלית מעל 80 ומדעי הרוח מעל 80 – לימודי פסיכולוגיה. ממוצע מעל 55 אנגלית מעל 80 ומדעי הרוח מעל 80 – לימודי פסיכולוגיה סוציולוגיה.

כל המקצעות מלבד מדעי הרוח מעל 80 – לימודי אלקטרוניקה.

אנגלית מעל 80, מדעי רוח מעל 80, כל השער מעל 55 – מעשייה וטיול.

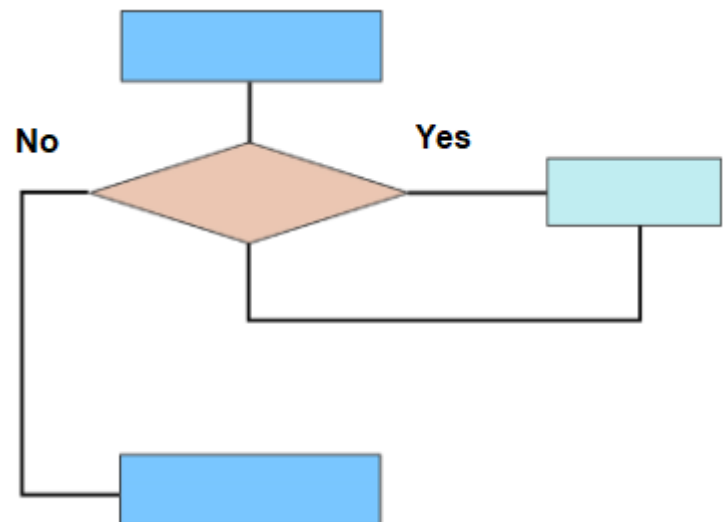
ממוצע מעל 60 ללא אף שלילי (מעל 55) – שיווק.

## פרק ד. ביצוע חוזר.

לולאות הן חלק חשוב בתכנות מבנית כמו ביצוע התניות, באופן זה ניתן לבצע חזרות עבור התניה מסוימת. באמצעות לולאות תוכל לארגן את החזרה על קטעי קוד. הצורך בכך עולה לעיתים קרובות למדי. לדוגמה, משתמש מזין מספרים ברצף, ויש להוסיף כל אחד מהם לסך הכול. או שיש להציג את הריבועים של סדרה של מספרים טבעיים וכדומה.

### לולאת while.

לולאת while היא לולאה שחוזרת על עצמה מספר פעמים. כל עוד ההתניה מתקיימת, קטע הקוד בלולאה יחזור על עצמו.



דומה לתנאי if. עם זאת, במקרה של מפעילים מחזוריים, ניתן לבצע את "גופם" לא פעם. במקרה של if, אם הביטוי הלוגי בכותרת מחזיר "אמת", אז הגוף מבוצע פעם אחת. לאחר מכן, זרימת ביצוע התוכנית חוזרת למסלול הראשי ומבצעת את הביטויים הבאים מתחת למשפט if. במקרה של while, לאחר ביצוע "גופו", הקוד חוזר לכותרת הלולאה ובודק שוב את הביטוי בתנאי. אם הביטוי הבוליאני חוזר לאמיתו, הגוף מתבצע שוב. ואז שוב אנו חוזרים לכותרת וכן הלאה. המחזור מסיים את עבודתו רק כאשר הביטוי הלוגי בכותרת מחזיר "שקר", כלומר, התנאי לביצוע המחזור כבר לא מתקיים. לאחר מכן זרימת הביצוע עוברת לביטויים שנמצאים מתחת למחזור כולו. אנו אומרים, "לצאת מהלולאה", כלומר להמשיך בתוכנית הראשית.

### שני חריגות ללולאת while:

- אם בכניסה הראשונה ללולאה, הביטוי הלוגי מחזיר "שקר", גוף הלולאה לא מבוצע אפילו פעם אחת. סיטואציה זו יכולה להיחשב כרגילה, מכיוון שבתנאים מסוימים לוגיקת התוכנית עשויה להניח כי אין צורך לבצע ביטויים של גוף הלולאה.
- אם הביטוי הבוליאני בכותרת ה- while לעולם לא מחזיר "שקר", אך תמיד נשאר "אמת", הלולאה לעולם לא תסתיים אלא אם כן קיימת הצהרת יציאה break בגופה, או שקוראת ליציאה מהפונקציות של התוכנית בעזרת exit (צא). אם המחזור חוזר וחוזר על עצמו מספר אינסופי של לולאות, התוכנית גורסת. בשלב זה הוא קופא ולא יכול להסתיים באופן עצמאי.



נשנה את הדוגמה מהסעיף הקודם כך, שתוכנית תעשה בדיקות כל עוד קלט לא יהיה תקין.

```
n = input("Enter the integer: ")
while type(n) != int:
    try:
        n = int(n)
    except ValueError:
        print("Illegal input!")
        n = input("Enter the integer: ")
if n % 2 == 0:
    print(n, " Is Even")
else:
    print(n, " Is Odd")
```

```
Enter the integer: a
Illegal input!
Enter the integer: 1.2
Illegal input!
Enter the integer: -2
-2 Is Even
```

דוגמה – חישוב סכום המספרים בין 1 לבין מספר הנקלט מהמשתמש.

```
num = int(input("Enter a number: "))
if num < 0:
    print("Enter a positive number")
else:
    sum = 0
    # use while loop to iterate until zero
    while(num > 0):
        sum += num
        num -= 1
    print("The sum is", sum)
```

```
Enter a number: 12
The sum is 78
Enter a number: 0
The sum is 0
Enter a number: -1
Enter a positive number
```

דוגמה – כתוב תוכנית שמבקשת מהמשתמש להכניס את החזקה ותוכנית תחשב 2 בחזקה זו.

```
n = input("Enter the integer: ")
answ=1
while type(n) != int:
    try:
        n = int(n)
    except ValueError:
        print("Illegal input!")
        n = input("Enter the integer: ")
while n!=0:
    answ=answ*2
    n=n-1
print(answ)
```

```
Enter the integer: 6
64
Enter the integer: 10
1024
```

שימו לב שהקוד הזה לא מושלם – אם ניתן מספר שלילי נקבל לולאה אין סופית:

```
Enter the integer: -1
-9
3
5
5
6
5
q
a
```

ננסה לתקן שגיאה זו:

```
n = input("Enter the integer: ")
answ=1
while type(n) != int:
    try:
        n = int(n)
    except ValueError:
        print("Illegal input!")
        n = input("Enter the integer: ")
if n>0:
    while n!=0:
```

```
    answ=answ*2
    n=n-1
    print(answ)
else: print("Enter positive number")
```

```
Enter the integer: -1
Enter positive number
Enter the integer: 10
1024
Enter the integer: a
Illegal input!
Enter the integer: -1
Enter positive number
```

דוגמה – בניית סדרת פיבונצ'י

(איבר ראשון – 0, איבר שני – 1, כל איבר הבא שווה לסכום של שני איברים הקודמים.)

```
# Program to display the Fibonacci sequence up to n-th term
nterms = int(input("How many terms? "))
# first two terms
n1, n2 = 0, 1
count = 0
# check if the number of terms is valid
if nterms <= 0:
    print("Please enter a positive integer")
elif nterms == 1:
    print("Fibonacci sequence upto",nterms,":")
    print(n1)
else:
    print("Fibonacci sequence:")
    while count < nterms:
        print(n1,end=" , ")
        nth = n1 + n2
        # update values
        n1 = n2
        n2 = nth
        count += 1
```

```
How many terms? 10
Fibonacci sequence:
0, 1, 1, 2, 3, 5, 8, 13, 21, 34,
```

דוגמה: בדיקה האם מספר הוא מספר "ארמסטרונג"?

מספר ארמסטרונג – מספר שאם ניקח את המספר ונפרק אותו לספרות, כל ספרה נעלה בחזקה של כמות הספרות ונחבר אותם, נקבל את המספר המקורי.

$$3*3*3 + 5*5*5 + 1*1*1 = 153$$

תוכנית לבדיקה של מספר תלת ספרתי:

```
# Python program to check if the number is an Armstrong number or not
# take input from the user
num = int(input("Enter a number: "))
# initialize sum
sum = 0
# find the sum of the cube of each digit
temp = num
while temp > 0:
    digit = temp % 10
    sum += digit ** 3
    temp //= 10
# display the result
if num == sum:
    print(num,"is an Armstrong number")
else:
    print(num,"is not an Armstrong number")
```

```
Enter a number: 153
153 is an Armstrong number
```

```
Enter a number: 121
121 is not an Armstrong number
```

קוד לחיפוש מספר ארמסטרונג מסדר כלשהו.

```
# Python program to check if the number is an Armstrong number or not
# take input from the user and calculated the length (number of digits)
# Changed num string to variable
num = input("Enter a number: ")
order = len(str(num))
num=int(num)
# initialize sum
sum = 0
# find the sum of the cube of each digit
temp = num
while temp > 0:
```

```

digit = temp % 10
sum += digit ** order
temp //= 10
# display the result
if num == sum:
    print(num,"is an Armstrong number")
else:
    print(num,"is not an Armstrong number")

```

```

Enter a number: 1634
1634 is an Armstrong number
Enter a number: 135
135 is not an Armstrong number
Enter a number: 407
407 is an Armstrong number

```

## לולאה Do While

לולאה ה- do while בשפת Python לא קיימת ומתבצעת ע"י לולאה while .

לולאה While שווי ערך בשפת Python	לולאה Do While בשפת C
<pre> a=10 while a&lt;20:     print(a)     a+=1 </pre>	<pre> #include &lt;stdio.h&gt; int main () {     int a = 10;     do {         printf("value of a: %d\n", a);         a = a + 1;     }while( a &lt; 20 );     return 0; } </pre>

## לולאת for

לולאת for יש שימוש, כאשר אנו יודעים כמות "איטרציות" (סיבובים, חזרות).

לולאות משמשות באופן מסורתי כשיש לך קטע קוד שעליו ברצוננו לחזור על מספר קבוע של פעמים.

לולאת ה- for היא כבר קצת יותר מסובכת, קצת פחות אוניברסלית, אך היא פועלת הרבה יותר מהר מאשר לולאת while. לולאה זו עוברת על אובייקט נתון הבנוי ממספר איברים (למשל, מחרוזת, רשימה, מערך), ובמהלך כל מעבר היא מבצעת את גוף הלולאה עבור חלק נבחר באובייקט, או עבור איבר נבחר במערך. החלק נבחר לפי כמות הפעמים שרצה הלולאה עד הרגע הנתון.

```

for i in 'hello world':
    print(i, ',')

```

```
h,  
e,  
l,  
l,  
o,  
,  
w,  
o,  
r,  
l,  
d,
```

```
for i in 'hello world':  
    print(i * 2, end="")
```

```
hheelllloo wwoorrlldd
```

## אופרטורים continue ו- break.

אופרטור continue מתחיל את הלולאה הבאה ללא סיום של הלולאה הנוכחית:

```
for i in 'hello world':  
    if i == 'o':  
        continue  
    print(i * 2, end="")
```

```
hheellll wwrrlldd
```

לעומת זאת אופרטור break מפסיק את ביצוע הלולאה הנוכחית ויוצא ממנה.

```
for i in 'hello world':  
    if i == 'o':  
        break  
    print(i * 2, end="")
```

```
hheellll
```

## אופרטור else בתוך לולאה:

המילה else, המשמשת בלולאת for או while, בודקת אם הצהרת break יצאה מהלולאה, או שהיציאה מהלולאה התבצעה בצורה "טבעית". הקוד שבתוך else יבוצע רק הקוד יצא מהלולאה ללא אופרטור .break

נראה את הדוגמה – אני רוצה להפסיק סריקת מחרוזת ברגע שמופיע אות "a".

צורה ראשונה:

```
for i in 'hello world':  
    if i == 'a':  
        break  
else:  
    print("There are no letter a in the sentence")
```

There are no letter a in the sentence

```
for i in 'hella world':  
    if i == 'a':  
        break  
else:  
    print("There are no letter a in the sentence")
```

```
C:\Users\marka\AppData\Local\Programs\Python\Python38-32\python.exe C:/MyPython/third.py  
Process finished with exit code 0
```

חשוב לשים לב שה-else צריך להיות מחוץ ללולאה:

```
for i in 'hello world':  
    if i == 'a':  
        break  
else:  
    print("There are no letter a in the sentence")
```

There are no letter a in the sentence  
There are no letter a in the sentence  
There are no letter a in the sentence  
There are no letter a in the sentence  
There are no letter a in the sentence  
There are no letter a in the sentence  
There are no letter a in the sentence  
There are no letter a in the sentence  
There are no letter a in the sentence  
There are no letter a in the sentence  
There are no letter a in the sentence

```
for i in 'hella world':
    if i == 'a':
        break
    else:
        print("There are no letter a in the sentence")
```

```
There are no letter a in the sentence
There are no letter a in the sentence
There are no letter a in the sentence
There are no letter a in the sentence
```

else עובד לא רק לאחר break, אלא גם ביציאה מהלולאה:

```
for x in range(3):
    print(x)
else:
    print('Final x = %d' % (x))
```

```
0
1
2
Final x = 2
```

## פונקציית range().

עכשיו הגיע הזמן להכיר את פונקציית range() המובנית בפייתון. "range()" מתורגם ל"טווח". זה יכול לקחת פרמטר אחד, שניים או שלושה. מטרתם זהה לפונקציה randrange() של המודול האקראי. אם מצוין רק מספר אחד, נוצר מערך של מספרים מ-0 למספר שצוין, לא כולל המספר שצוין. אם ניתנים שניים, נוצר מערך של מספרים מהמספר הראשון לשני, לא כולל המספר השני. אם ניתנים שלושה, נוצר מערך של מספרים מהמספר הראשון לשני לא כולל המספר השני, כשהמספר השלישי מצוין את מספר הקפיצות שצריך להיות בין כל איבר במערך.

עם זאת, שלא כמו randrange(), פונקציית range() מייצרת יותר ממספר אקראי אחד בטווח מוגדר, ובכלל לא מייצרת מספר אקראי. פונקציה זו מייצרת רצף של מספרים בטווח שצוין. אז range(5, 11) יפיק רצף של 5, 6, 7, 8, 9, 10. עם זאת, זה לא יהיה מבנה נתונים מסוג רשימה.

פונקציית range() מייצרת אובייקטים מהמחלקה שלה - טווחים:

```
>>> a=range(10)
>>> a
range(0, 10)
>>> a=range(-2,2)
>>> a
range(-2, 2)
>>> a=range(-2,10,2)
```



```
>>> type(a)
<class 'range'>
```

כדי לגשת לאיברים של a יש לציין מיקומם במערך (נושא הזה נלמד בהמשך).

```
>>> a[0]
-2
>>> a[1]
0
>>> a[2]
2
```

מושג "איטרטורים" - Iterators.

בפייתון "איטרטור" - הוא כל מערך או מבנה נתונים שניתן להשתמש במצביע להצביע לאיבר במערך שעליו פועלת הלולאה for. רשימות, צירופים, מילונים וסטים הם כולם דוגמאות לאיטרטורים מובנים. סוגים אלה הם איטרטורים מכיוון שהם מיישמים את השיטות הבאות. למעשה, כל אובייקט שרוצה להיות איטרטור חייב ליישם את השיטות הבאות.

סוג א'. שיטת `__iter__` - שיטה הנקראת אתחול איטרטור. זה אמור להחזיר אובייקט שיש לו שיטה `__next__`.

סוג ב'. `__next__` השיטה הבאה של איטרטור צריכה להחזיר את הערך הבא עבור ערך החוזר. כאשר משתמשים באיטרטור עם לולאה 'for in', הלולאה for מתקשרת באופן מרומז `next()` על אובייקט האיטרטור. שיטה זו צריכה להעלות `StopIteration` כדי לאותת על סיום האיטרציה.

```
# Here is an example of a python inbuilt iterator
# value can be anything which can be iterate
iterable_value = "Hello"
iterable_obj = iter(iterable_value)
while True:
    try:
        # Iterate by calling next
        item = next(iterable_obj)
        print(item)
    except StopIteration:
        # exception will happen when iteration will over
        Break
```

```
H
e
l
l
o
```

דוגמה של שימוש בלולאת for ופונקציה range():  
עבור לולאה מ 0 ל 2, ולכן פועל 3 פעמים.

```
>>> for i in range(10):
    print(i, end=" ")
0 1 2 3 4 5 6 7 8 9
>>> for i in range(-10,10):
    print(i, end=" ")
-10 -9 -8 -7 -6 -5 -4 -3 -2 -1 0 1 2 3 4 5 6 7 8 9
>>> for i in range(-10,10,2):
    print(i, end=" ")
-10 -8 -6 -4 -2 0 2 4 6 8
```

### פונקציה enumerate():

לעומת פונקציה range() שמקבלת רק מיקום של איברים, פונקציה enumerate() מחזירה זוגות של איברים, כאשר במקום ראשון עומד מיקום האיבר ובמקום השני איבר עצמו.

```
>>> reshima=[2,5,1,2,3,5]
>>> for i in enumerate (reshima):
    print(i)

(0, 2)
(1, 5)
(2, 1)
(3, 2)
(4, 3)
(5, 5)
```

### לולאה בתוך לולאה – for בתוך for.

```
for x in range(1, 11):
    for y in range(1, 11):
        if y==10:
            print('%d * %d = %d ' % (x, y, x * y))
        else:
            print('%d * %d = %d ' % (x, y, x*y),end=" ")
```

```
1 * 1 = 1 1 * 2 = 2 1 * 3 = 3 1 * 4 = 4 1 * 5 = 5 1 * 6 = 6 1 * 7 = 7 1 * 8 = 8 1 * 9 = 9 1 *
10 = 10
2 * 1 = 2 2 * 2 = 4 2 * 3 = 6 2 * 4 = 8 2 * 5 = 10 2 * 6 = 12 2 * 7 = 14 2 * 8 = 16 2 * 9 = 18
2 * 10 = 20
3 * 1 = 3 3 * 2 = 6 3 * 3 = 9 3 * 4 = 12 3 * 5 = 15 3 * 6 = 18 3 * 7 = 21 3 * 8 = 24 3 * 9 =
27 3 * 10 = 30
```

$4 * 1 = 4$   $4 * 2 = 8$   $4 * 3 = 12$   $4 * 4 = 16$   $4 * 5 = 20$   $4 * 6 = 24$   $4 * 7 = 28$   $4 * 8 = 32$   $4 * 9 = 36$   $4 * 10 = 40$   
 $5 * 1 = 5$   $5 * 2 = 10$   $5 * 3 = 15$   $5 * 4 = 20$   $5 * 5 = 25$   $5 * 6 = 30$   $5 * 7 = 35$   $5 * 8 = 40$   $5 * 9 = 45$   $5 * 10 = 50$   
 $6 * 1 = 6$   $6 * 2 = 12$   $6 * 3 = 18$   $6 * 4 = 24$   $6 * 5 = 30$   $6 * 6 = 36$   $6 * 7 = 42$   $6 * 8 = 48$   $6 * 9 = 54$   $6 * 10 = 60$   
 $7 * 1 = 7$   $7 * 2 = 14$   $7 * 3 = 21$   $7 * 4 = 28$   $7 * 5 = 35$   $7 * 6 = 42$   $7 * 7 = 49$   $7 * 8 = 56$   $7 * 9 = 63$   $7 * 10 = 70$   
 $8 * 1 = 8$   $8 * 2 = 16$   $8 * 3 = 24$   $8 * 4 = 32$   $8 * 5 = 40$   $8 * 6 = 48$   $8 * 7 = 56$   $8 * 8 = 64$   $8 * 9 = 72$   $8 * 10 = 80$   
 $9 * 1 = 9$   $9 * 2 = 18$   $9 * 3 = 27$   $9 * 4 = 36$   $9 * 5 = 45$   $9 * 6 = 54$   $9 * 7 = 63$   $9 * 8 = 72$   $9 * 9 = 81$   $9 * 10 = 90$   
 $10 * 1 = 10$   $10 * 2 = 20$   $10 * 3 = 30$   $10 * 4 = 40$   $10 * 5 = 50$   $10 * 6 = 60$   $10 * 7 = 70$   $10 * 8 = 80$   $10 * 9 = 90$   $10 * 10 = 100$

דוגמה: בדיקה האם מספר שמשתמש מכניס הוא ראשוני או לא.

```

# Program to check if a number is prime or not
# To take input from the user
num = int(input("Enter a number: "))

# prime numbers are greater than 1
if num > 1:
    # check for factors
    for i in range(2, num):
        if (num % i) == 0:
            print(num, "is not a prime number")
            print(i, "times", num // i, "is", num)
            break
    else:
        print(num, "is a prime number")

# if input number is less than
# or equal to 1, it is not prime
else:
    print(num, "is not a prime number")

```

Enter a number: 125  
125 is not a prime number  
5 times 25 is 125  
  
Enter a number: 3  
3 is a prime number

```
Enter a number: -6
-6 is not a prime number
```

דוגמה: תוכנית שמדפיסה כל המספרים הראשוניים בטווח שנתון ע"י המשתמש. (יש לבדוק תקינות הקלט).

```
# Python program to display all the prime numbers within an interval
lower = int(input("Enter a low limit: "))
upper = int(input("Enter an upper limit: "))
if lower>0 and upper>0:
    if upper<lower:
        temp=lower
        lower=upper
        upper=temp
    print("Prime numbers between", lower, "and", upper, "are:")
    for num in range(lower, upper + 1):
        # all prime numbers are greater than 1
        if num > 1:
            for i in range(2, num):
                if (num % i) == 0:
                    break
            else:
                print(num, end=" ")
else: print("You have to enter only positive numbers")
```

```
Enter a low limit: 35
Enter an upper limit: 125
Prime numbers between 35 and 125 are:
37, 41, 43, 47, 53, 59, 61, 67, 71, 73, 79, 83, 89, 97, 101, 103, 107, 109, 113,

Enter a low limit: 125
Enter an upper limit: 35
Prime numbers between 35 and 125 are:
37, 41, 43, 47, 53, 59, 61, 67, 71, 73, 79, 83, 89, 97, 101, 103, 107, 109, 113,

Enter a low limit: -9
Enter an upper limit: 12
You have to enter only positive numbers
```

דוגמה למציאת n!. (1 \* 2 \* 3 \* ... \* n).

```
# Python program to find the factorial of a number provided by the user.
# To take input from the user
num = int(input("Enter a number: "))
factorial = 1
# check if the number is negative, positive or zero
if num < 0:
    print("Sorry, factorial does not exist for negative numbers")
elif num == 0:
    print("The factorial of 0 is 1")
else:
    for i in range(1,num + 1):
        factorial = factorial*i
    print("The factorial of",num,"is",factorial)
```

```
Enter a number: -9
Sorry, factorial does not exist for negative numbers

Enter a number: 0
The factorial of 0 is 1

Enter a number: 6
The factorial of 6 is 720
```

דוגמה – בניית סדרת פיבונצ'י (איבר ראשון – 0, איבר שני – 1, כל איבר הבא שווה לסכום של שני איברים הקודמים).

```
# Program to display the Fibonacci sequence up to n-th term
nterms = int(input("How many terms? "))
# first two terms
n1, n2 = 0, 1
count = 0
# check if the number of terms is valid
if nterms <= 0:
    print("Please enter a positive integer")
elif nterms == 1:
    print("Fibonacci sequence upto",nterms,":")
    print(n1)
else:
    print("Fibonacci sequence:")
    for count in range(1,nterms+1):
        print(n1,end=" , ")
        nth = n1 + n2
```

```
# update values
```

```
n1 = n2
```

```
n2 = nth
```

How many terms? 10

Fibonacci sequence:

0, 1, 1, 2, 3, 5, 8, 13, 21, 34,

מצא את מספרי ארמסטרונג בתחום הנקלט מהמשתמש.

```
lower = int(input("Enter a low limit: "))
upper = int(input("Enter an upper limit: "))
if lower > 0 and upper > 0:
    if upper < lower:
        temp = lower
        lower = upper
        upper = temp
    print("Prime numbers between", lower, "and", upper, "are:")
    for num in range(lower, upper + 1):
        # order of number
        order = len(str(num))
        # initialize sum
        sum = 0

        temp = num
        while temp > 0:
            digit = temp % 10
            sum += digit ** order
            temp //= 10
        if num == sum:
            print(num, end=" ")
else: print("You have to enter two positive numbers.")
```

Enter a low limit: 100  
Enter an upper limit: 300  
Prime numbers between 100 and 300 are:  
153

Enter a low limit: 1000  
Enter an upper limit: 200  
Prime numbers between 200 and 1000 are:  
370, 371, 407,

Enter a low limit: 200  
Enter an upper limit: -100  
You have to enter two positive numbers.

טסליצקי מרק

תרגול בנושא ביצוע חוזר:  
תרגול בנושא טבלאות מעקב.  
תרגיל מס' 1.

תפענח את הקוד הבא והצג מספר דוגמאות של פלט עבור קלט שונה.

```
import random
for i in range(0,5):
    x,y = [random.randrange(-3, 4) for i in range(0,2)]
    print("\n Point (x,y): ({0},{1}).format(x,y))
    if x == 0 and y == 0:
        print("0. Center")
    elif y == 0:
        print("1. Line OX")
    elif x == 0:
        print("2. Line OY")
    else:
        print("3. Not on the axis")
```

תרגיל מס' 2.

תפענח את הקוד הבא והצג מספר דוגמאות של פלט עבור קלט שונה.

```
import math
for i in range(-5,6):
    h = i/2
    x = math.pi * h
    if x > 0:
        y = 2*math.sin(x)
    else:
        y = 6 - x
    print("h = {0} : x = {1:.4f} : f(x) = {2:.4f}".format(h,x,y))
```

תרגיל מס' 3.

תפענח את הקוד הבא והצג מספר דוגמאות של פלט עבור קלט שונה.

```
for x in range(-4,5):
    if x < -2 or x > 2:
        y = 2*x
    else:
        y = -3 * x
    print("x = {0} : f(x) = {1}".format(x,y))
```



תרגיל מס' 4.

תפענח את הקוד הבא והצג מספר דוגמאות של פלט עבור קלט שונה.

```
for i in range(-2,7):
    x = i / 2
    if x <= 0:
        y = -x
    elif x < 2:
        y = x * x
    else:
        y = 4
    print("x = {0} : f(x) = {1}".format(x,y))
```

תרגיל מס' 5.

תפענח את הקוד הבא והצג מספר דוגמאות של פלט עבור קלט שונה.

```
import math
x = -1
while x < 11:
    x_floor = math.floor(x)
    if x < 0:
        y = 0
    elif x_floor%2 == 0:
        y = 1
    else:
        y = -1
    print("x = {0} : f(x) = {1}".format(x,y))
    x += .5
```

תרגיל מס' 6.

תפענח את הקוד הבא והצג מספר דוגמאות של פלט עבור קלט שונה.

```
import random
N = random.randrange(10)+1
K = random.randrange(-10, 10)
print('K = ', K)
print('N = ', N)
for i in range(0, N):
    print(i+1, " : ", K)
```

תרגיל מס' 7.

תפענח את הקוד הבא והצג מספר דוגמאות של פלט עבור קלט שונה.

```
for i in range(-4,5):
    if i == 0:
        s = "zero "
    elif i > 0:
        s = "positive "
    else:
        s = "negative "
    if i != 0:
        if i%2 == 0:
            s += "even "
        else:
            s += "odd "
    s += "number"
    print(i, " : ",s)
```

תרגיל מס' 8.

תפענח את הקוד הבא והצג מספר דוגמאות של פלט עבור קלט שונה.

```
import random
A = random.randrange(10)
n = random.randrange(10)+1
B = A + n
print('A = ', A)
print('B = ', B)
N = 0
for i in range(A+1,B,1):
    N +=1
    print(i, " : ",N)
print("N = ",N)
```

תרגיל מס' 9.

תפענח את הקוד הבא והצג מספר דוגמאות של פלט עבור קלט שונה.

```
import random
B = random.randrange(2,20)
A = random.randrange(1,B)
print('A = ', A)
print('B = ', B)
N = 0
for i in range(B-1, A, -1):
```

```
print(i,end=' ')
N += 1
print()
print("N = ", N)
```

תרגיל מס' 10.

תפענח את הקוד הבא והצג מספר דוגמאות של פלט עבור קלט שונה.

```
import random
A = random.randrange(10)
n = random.randrange(10)+1
B = A + n
print('A = ', A)
print('B = ', B)
S = 0
for i in range(A,B+1,1):
    S += i
    print(i, " : ", S)
print("Sum = ", S)
```

תרגיל מס' 11.

תפענח את הקוד הבא והצג מספר דוגמאות של פלט עבור קלט שונה.

```
import random
A = random.randrange(10)
n = random.randrange(10)+1
B = A + n
print('A = ', A)
print('B = ', B)
P = 1
for i in range(A,B+1,1):
    P *= i
    print(i, " : ", P)
print("Product = ", P)
```

תרגיל מס' 12.

תפענח את הקוד הבא והצג מספר דוגמאות של פלט עבור קלט שונה.

```
import random
A = random.randrange(10)
n = random.randrange(10)+1
B = A + n
```

```

print('A = ', A)
print('B = ', B)
S = 0
for i in range(A,B+1,1):
    S += i*i
    print(i, " : ",i*i," : ",S)
print("Sum of squares = ",S)

```

תרגיל מס' 13.

תפענח את הקוד הבא והצג מספר דוגמאות של פלט עבור קלט שונה.

```

import random
N = random.randrange(10)
print('N = ', N)
S = 0.0
for i in range(N):
    S += 1/(i+1)
    print(i+1, " : ",1/(i+1)," : ",S)
print("Sum = ",S)

```

תרגיל מס' 14.

תפענח את הקוד הבא והצג מספר דוגמאות של פלט עבור קלט שונה.

```

import random
N = random.randrange(10)
print('N = ', N)
S = 0.0
for i in range(N,2*N+1):
    x = i**2
    S += x
    print(i, " : ",x," : ",S)
print("Sum = ",S)

```

תרגיל מס' 15.

תפענח את הקוד הבא והצג מספר דוגמאות של פלט עבור קלט שונה.

```

import random
N = random.randrange(1,10)
print('N = ', N)
P = 1.0

```

```

for i in range(1,N+1):
    x = 1 + i*0.1
    P *= x
    print(i, " : ",x, " : ",P)
print("Product = ",P)

```

תרגיל מס' 16.

תפענח את הקוד הבא והצג מספר דוגמאות של פלט עבור קלט שונה.

```

import random
N = random.randrange(1,15)
print('N = ', N)
S = 0.0
for i in range(1,N+1):
    x = (1 + i*0.1)*(-1)**(i+1)
    S += x
    print(i, " : ",x, " : ",S)
print("Sum = ",S)

```

תרגיל מס' 17.

תפענח את הקוד הבא והצג מספר דוגמאות של פלט עבור קלט שונה.

```

import random
N = random.randrange(1,15)
print('N = ', N)
S = 0.0
for i in range(1,N+1):
    x = 2 * i - 1
    S += x
    print(i, " : ",x, " : ",S)
print("Sum = ",S)

```

תרגיל מס' 18.

תפענח את הקוד הבא והצג מספר דוגמאות של פלט עבור קלט שונה.

```

import random
N = random.randrange(1,10)
print('N = ', N)
A = random.randrange(-10,10)
print('A = ', A)
P = 1
for i in range(1,N+1):

```

```
P *= A
print(i, " : ", P)
print("Product = ", P)
```

תרגיל מס' 19.

תפענח את הקוד הבא והצג מספר דוגמאות של פלט עבור קלט שונה.

```
import random
N = random.randrange(1,10)
print('N = ', N)
A = random.randrange(-10,10)
print('A = ', A)
P = 1
for i in range(1,N+1):
    P *= A
    print(A, " power ", i, " equal", P)
```

תרגיל מס' 20.

תפענח את הקוד הבא והצג מספר דוגמאות של פלט עבור קלט שונה.

```
import random
N = random.randrange(1,10)
print('N = ', N)
A = random.randrange(-10,10)
print('A = ', A)
P = 1.0
S = 1.0
for i in range(1,N+1):
    P *= A
    S += P
    print(i, " : ", P, " : ", S)
print("Result:", S)
```

תרגיל מס' 21.

תפענח את הקוד הבא והצג מספר דוגמאות של פלט עבור קלט שונה.

```
import random
N = random.randrange(1,10)
print('N = ', N)
A = random.randrange(-10,10)
print('A = ', A)
P = 1.0
S = 1.0
```

```

for i in range(1,N+1):
    P *= A * (-1)
    S += P
    print(i, " : ", P, " : ", S)
print("Result:",S)

```

תרגיל מס' 22.

תפענח את הקוד הבא והצג מספר דוגמאות של פלט עבור קלט שונה.

```

import random
N = random.randrange(1,15)
print('N = ', N)
F = 1.0
for i in range(1,N+1):
    F *= i
    print(i, " : ", F)
print("Result:",F)

```

תרגיל מס' 23.

תפענח את הקוד הבא והצג מספר דוגמאות של פלט עבור קלט שונה.

```

import random
N = random.randrange(1,10)
print('N = ', N)
F = 1.0
S = 0.0
for i in range(1,N+1):
    F *= i
    S += F
    print(i, " : ", F, " : ", S)
print("Result:",S)

```

תרגיל מס' 24.

תפענח את הקוד הבא והצג מספר דוגמאות של פלט עבור קלט שונה.

```

import random
import math
N = random.randrange(1,15)
N = 20
print('N = ', N)
F = 1.0
S = 1.0

```

```

for i in range(1,N+1):
    F /= i
    S += F
    print(i, " : ", F, " : ", S)
print("Result:",S)
print("e = ",math.exp(1))

```

תרגיל מס' 25.

תפענח את הקוד הבא והצג מה פלט התוכנה.

```

import math
X = 2
N = 20
print('X = ', X)
print('N = ', N)
F = 1.0
S = 1.0
for i in range(1,N+1):
    F *= X / i
    S += F
    print(i, " : ", F, " : ", S)
print("Result:")
print(S)
print("e:")
print(math.exp(X))

```

תרגיל מס' 26.

תפענח את הקוד הבא והצג מה פלט התוכנה.

```

import math
X = math.pi/4
N = 10
print('X = ', X)
print('N = ', N)
p = X
S = X
k = 0
for i in range(1,N+1):
    k += 2
    p *= (-1) * X * X / (k * (k + 1))
    S += p
    print(i, " : ", k, " : ", k+1, " : ", p, " : ", S)
print("Result:")

```



```
print(S)
print("sin:")
print(math.sin(X))
```

תרגיל מס' 27.

תפענח את הקוד הבא והצג מה פלט התוכנה.

```
import math
X = math.pi/3
N = 10
print('X = ', X)
print('N = ', N)
p = 1
S = 1
k = 1
for i in range(1,N+1):
    p *= (-1) * X * X / (k * (k + 1))
    S += p
    print(i, " : ", k, " : ", k+1, " : ", p, " : ", S)
    k += 2
print("Result:")
print(S)
print("cos:")
print(math.cos(X))
```

תרגיל מס' 28.

תפענח את הקוד הבא והצג מה פלט התוכנה.

```
import math
X = 0.5
N = 10
print('X = ', X)
print('N = ', N)
p = 1
S = 0
for i in range(1,N+1):
    p *= X
    S += p/i
    print(i, " : ", p, " : ", S)
    p *= -1
print("Result:")
print(S)
print("ln(x+1):")
print(math.log(X+1))
```

תרגיל מס' 29.

תפענח את הקוד הבא והצג מה פלט התוכנה.

```
import math
X = 0.8
N = 20
print('X = ', X)
print('N = ', N)
p = X
S = X
k = 1
for i in range(1,N+1):
    p *= -1*X*X
    k += 2
    S += p/k
    print(i, " : ", k, " : ", p, " : ", S)
print("Result:")
print(S)
print("atan(x):")
print(math.atan(X))
```

תרגיל מס' 30.

תפענח את הקוד הבא והצג מה פלט התוכנה.

```
import math
X = 0.9
N = 100
print('X = ', X)
print('N = ', N)
p = X
S = X
k = 1
for i in range(1,N+1):
    p *= k/((k+1)*(k+2)) * X*X
    S += p
    print(i, " : ", k, " : ", p, " : ", S)
    k += 2
    p *= k
print("Result:")
print(S)
print("asin(x):")
print(math.asin(X))
```

תרגיל מס' 31.

תפענח את הקוד הבא והצג מה פלט התוכנה.

```
import math
X = -0.9
N = 100
print('X = ', X)
print('N = ', N)
p = X/2
S = 1 + X/2
k = 2
for i in range(2,N+1):
    k += 2
    p *= (-1)*(k-3)/k * X
    S += p
    print(i, " : ", k, " : ", p, " : ", S)
print("Result:")
print(S)
print("sqrt(1+x):")
print(math.sqrt(1+X))
```

תרגיל מס' 32.

תפענח את הקוד הבא והצג מה פלט התוכנה.

```
import random
B = random.randrange(-9,11)
A = random.randrange(-10,B)
N = random.randrange(2,11)
H = (B - A) / N
print("A = ",A)
print("B = ",B)
print("N = ",N)
print("H = ",H)
x = A
for i in range(0,N):
    print(x,end=" ")
    x += H
print(x)
```

תרגיל מס' 33.

תפענח את הקוד הבא והצג מספר דוגמאות של פלט עבור קלט שונה.

```
import random
import math
B = random.randrange(-9,11)
A = random.randrange(-10,B)
N = random.randrange(2,11)
H = (B - A) / N
print("A = ",A)
print("B = ",B)
print("N = ",N)
print("H = {0:.2f}".format(H))
x = A
for i in range(0,N+1):
    y = 1 - math.sin(x)
    print("{0:.2f} : {1:.4f}".format(x,y))
    x += H
```

תרגיל מס' 34.

תפענח את הקוד הבא והצג מה פלט התוכנה.

```
for N in range(1,6,2):
    print("N = ",N)
    A0 = 2
    for k in range(1,N+1):
        A1 = 2 + 1/A0
        print(k, " : ",A1)
        A0 = A1
```

תרגיל מס' 35.

תפענח את הקוד הבא והצג מה פלט התוכנה.

```
for N in range(1,7,2):
    print("N = ",N)
    A0 = 1
    for k in range(1,N+1):
        A1 = (A0 + 1)/k
        print(k, " : ",A1)
        A0 = A1
```

תרגיל מס' 36.

תפענח את הקוד הבא והצג מה פלט התוכנה.

```
for N in range(2,7,2):
    print("N = ",N)
    F1 = 1
    F2 = 1
    print(1, " : ",F1)
    print(2, " : ",F2)
    for k in range(3,N+1):
        F3 = F2 + F1
        print(k, " : ",F3)
        F1 = F2
        F2 = F3
    print()
```

תרגיל מס' 37.

תפענח את הקוד הבא והצג מה פלט התוכנה.

```
for N in range(2,8,2):
    print("N = ",N)
    A1 = 1
    A2 = 2
    print(1, " : ",A1)
    print(2, " : ",A2)
    for k in range(3,N+1):
        A3 = (A1 + 2*A2)/3
        print(k, " : ",A3)
        A1 = A2
        A2 = A3
    print()
```

תרגיל מס' 38.

תפענח את הקוד הבא והצג מה פלט התוכנה.

```
for N in range(2,9,2):
    print("N = ",N)
    A1 = 1
    A2 = 2
    A3 = 3
    print(1, " : ",A1)
    print(2, " : ",A2)
```

```

print(3, " : ",A3)
for k in range(4,N+1):
    A4 = A3 + A2 - 2*A1
    print(k, " : ",A4)
    A1 = A2
    A2 = A3
    A3 = A4
print()

```

תרגיל מס' 39.

תפענח את הקוד הבא והצג מספר דוגמאות של פלט עבור קלט שונה.

```

import random
N = random.randrange(1,11)
K = random.randrange(1,11)
print("N = ",N)
print("K = ",K)
print()
s = 0
for i in range(1,N+1):
    p = 1.0
    for j in range(1,K+1):
        p *= i
    s += p
    print("i = ",i)
    print("p = ",p)
    print("s = ",s)
    print()
print("Sum = ",s)

```

תרגיל מס' 40.

תפענח את הקוד הבא והצג מספר דוגמאות של פלט עבור קלט שונה.

```

import random
N = random.randrange(1,11)
print("N = ",N)
print()
s = 0
for i in range(1,N+1):
    p = 1.0
    for j in range(1,i+1):
        p *= j
    s += p

```

```
print("i = ",i)
print("p = ",p)
print("s = ",s)
print()
print("Sum = ",s)
```

תרגיל מס' 41.

תפענח את הקוד הבא והצג מספר דוגמאות של פלט עבור קלט שונה.

```
import random
N = random.randrange(1,11)
print("N = ",N)
print()
s = 0
for i in range(1,N+1):
    p = 1.0
    K = N + 1 - i
    for j in range(0,K):
        p *= i
    s += p
    print("i = ",i)
    print("j = ",j)
    print("p = ",p)
    print("s = ",s)
    print()
print("Sum = ",s)
```

תרגיל מס' 42.

תפענח את הקוד הבא והצג מספר דוגמאות של פלט עבור קלט שונה.

```
import random
A = random.randrange(1,11)
B = random.randrange(A+1,21)
print("A = ",A)
print("B = ",B)
print()
for i in range(A,B+1):
    for j in range(0,i):
        print(i,end=" ")
    print()
```

תרגיל מס' 43.

תפענח את הקוד הבא והצג מספר דוגמאות של פלט עבור קלט שונה.

```
import random
A = random.randrange(50,100)
B = random.randrange(1,A)
print('A = ', A)
print('B = ', B)
r = A - B
while r >= B:
    r -= B
print("Remainder: ", r)
```

תרגיל מס' 44.

תפענח את הקוד הבא והצג מספר דוגמאות של פלט עבור קלט שונה.

```
import random
A = random.randrange(50,100)
B = random.randrange(1,A)
print('A = ', A)
print('B = ', B)
r = A - B
n = 1
while r >= B:
    r -= B
    n += 1
print(" B in A: ", n)
print("Remainder: ", r)
```

תרגיל מס' 45.

תפענח את הקוד הבא והצג מספר דוגמאות של פלט עבור קלט 42, 35, 27, 81.

```
N=int(input("Enter the num: "))
print('N = ', N)
while N >= 3:
    N /= 3
print("Really ", (N==1))
```



תרגיל מס' 46.

תפענח את הקוד הבא והצג מספר דוגמאות של פלט עבור קלט שונה.

```
import random
K = random.randrange(1,30)
N = 2 ** K
print('N = ', N)
K_new = 0
while N >= 2:
    N /= 2
    K_new += 1
print("K = ", K_new)
```

תרגיל מס' 47.

תפענח את הקוד הבא והצג מספר דוגמאות של פלט עבור קלט 3,5,7?

```
N = int(input("Enter the number from 3 to 10"))
print('N = ', N)
if N % 2 == 0:
    L = 2
else:
    L = 1
F = 1
while N >= L:
    F *= N
    N -= 2
print("Answer is: ", F)
```

תרגיל מס' 48.

תפענח את הקוד הבא והצג מספר דוגמאות של פלט עבור קלט שונה.

```
import random
N = random.randrange(100, 1000)
print('N = ', N)
K = 1
while K * K <= N:
    K += 1
print("K = {0}, max = {1}, min = {2}".format(K, K ** 2, (K - 1) ** 2))
```

תרגיל מס' 49.

תפענח את הקוד הבא והצג מספר דוגמאות של פלט עבור קלט שונה.

```
import random
N = random.randrange(100,1000)
print('N = ', N)
K = 0
P = 1
while P <= N:
    P *= 3
    K += 1
print("K = {0}, max = {1}, min = {2}".format(K,3**K,3**(K-1)))
```

תרגיל מס' 50.

תפענח את הקוד הבא והצג מספר דוגמאות של פלט עבור קלט שונה.

```
import random
N = random.randrange(5,10)
print('N = ', N)
K = 1
S = 1
while S < N:
    K += 1
    S += K
    print("K = {0}, S = {1}".format(K,S))
print()
print("K = {0}, S = {1}".format(K,S))
```

תרגיל מס' 51.

תפענח את הקוד הבא והצג מספר דוגמאות של פלט עבור קלט שונה.

```
import random
A = random.randrange(2,6)
print('A = ', A)
K = 1.0
S = 1.0
while S <= A:
    K += 1
    x = 1/K
    S += 1/K
    print("K = {0}, 1/K = {1}, S = {2}".format(K,x,S))
print()
print("K = {0}, S = {1}".format(K,S))
```

תרגיל מס' 52.

תפענח את הקוד הבא והצג מספר דוגמאות של פלט עבור קלט שונה.

```
import random
Daily = 10
Total = 200
P = random.randrange(20,51)
print('P = ', P)
coef = 1 + P/100
print("First Day = {0}, Percent = {1}, Coef = {2}".format(Daily,P,coef))
K = 1
S = Daily
while S < Total:
    Daily *= coef
    S += Daily
    K += 1
    print("K = {0}, Daily = {1}, S = {2}".format(K,Daily,S))
print()
print("Days = {0}, Summary = {1}".format(K,S))
```

תרגיל מס' 53.

תפענח את הקוד הבא והצג מספר דוגמאות של פלט עבור קלט שונה.

```
import random
N = random.randrange(1000,100000)
print("N = ",N)
q = N
while q >= 1:
    r = q % 10
    print(r,end="; ")
    q = int(q/10)
```

תרגיל מס' 54.

תפענח את הקוד הבא והצג מספר דוגמאות של פלט עבור קלט שונה.

```
import random
N = random.randrange(100,10000000)
print("N = ",N)
q = N
i = 0
s = 0
while q >= 1:
    i += 1
    r = q % 10
    s += r
    print(i, " - ",r, " - ",s)
    q = int(q/10)
print("Num:",i)
print("Sum:",s)
```

תרגיל מס' 54.

תפענח את הקוד הבא והצג מספר דוגמאות של פלט עבור קלט שונה.

```
import random
N = random.randrange(1,10000)
print("N = ",N)
q = N
s = 0
while q >= 1:
    r = q % 10
    q = int(q/10)
    s = s*10 + r
print("New Num:",s)
```

תרגיל מס' 55.

תפענח את הקוד הבא והצג מספר דוגמאות של פלט עבור קלט שונה.

```
import random
N = random.randrange(1,100000)
print("N = ",N)
q = N
i = 0
flag = False
while q >= 1:
```

```

i += 1
r = q % 10
print(i, " - ", r)
if r == 2:
    flag = True
    break
q = int(q/10)
print(flag)

```

תרגיל מס' 56.

תפענח את הקוד הבא והצג מספר דוגמאות של פלט עבור קלט שונה.

```

N = random.randrange(10,1000)
print("N = ",N)
q = N
i = 0
flag = False
while q >= 1:
    i += 1
    r = q % 10
    print(i, " - ", r)
    if r%2 == 1:
        flag = True
        break
    q = int(q/10)
print(flag)

```

תרגיל מס' 57.

תפענח את הקוד הבא והצג מספר דוגמאות של פלט עבור קלט שונה.

```

import math
N = random.randrange(2,100)
print("N = ",N)
for x in range(2,N+1):
    n = math.sqrt(x)
    i = 2
    k = 0
    while i <= n:
        if int(x / i)*i == x:
            k += 1
            break
        i += 1
    if k == 0:
        print(x,end="; ")

```

תרגיל מס' 58.

תפענח את הקוד הבא והצג מספר דוגמאות של פלט עבור קלט שונה.

```
import random
A = random.randrange(10,1000)
B = random.randrange(10,1000)
print("A = {0}, B = {1}".format(A,B))
while B > 0:
    A,B = B,A%B
print("Answer is: ",A)
```

תרגיל מס' 59.

תפענח את הקוד הבא והצג מספר דוגמאות של פלט עבור קלט שונה.

```
import random
N = random.randrange(10,100000)
print("N = ",N)
F1 = F2 = 1
print(F1,end="; ")
print(F2,end="; ")
while F2 < N:
    F1, F2 = F2, F1+F2
    print(F2,end="; ")
print()
if F2 == N:
    print("TRUE")
else:
    print("FALSE")
```

תרגיל מס' 60.

תפענח את הקוד הבא והצג מספר דוגמאות של פלט עבור קלט שונה.

```
import random
N = random.randrange(10,100000)
print("N = ",N)
F1 = F2 = 1
print(F1,end="; ")
print(F2,end="; ")
while F2 <= N:
    F1, F2 = F2, F1+F2
    print(F2,end="; ")
print()
print(F2)
```

תרגיל מס' 61.

תפענח את הקוד הבא והצג מספר דוגמאות של פלט עבור קלט שונה.

```
import random
N = random.randrange(1,15)
eps = 1/10**N
print("N = ",N)
print("eps = ",eps)
A1 = 2
print(1,":",A1)
A2 = 2 + 1/A1
k = 2
print(2,":",A2)
while abs(A2 - A1) >= eps:
    A1 = A2
    A2 = 2 + 1/A2
    k += 1
    print(k,":",A2)
print(k,":",A1,":",A2)
```

תרגיל מס' 62.

תפענח את הקוד הבא והצג מספר דוגמאות של פלט עבור קלט שונה.

```
import random
N = random.randrange(1,15)
eps = 1/10**N
print("N = ",N)
print("eps = ",eps)
A1 = 1
A2 = 2
A3 = (A1 + 2*A2)/3
print(1,":",A1)
print(2,":",A2)
print(3,":",A3)
k = 3
while abs(A3 - A2) >= eps:
    A2, A3 = A3, (A2 + 2*A3)/3
    k += 1
    print(k,":",A3)
print()
print(k,":",A2,":",A3)
```

תרגיל מס' 63.

תפענח את הקוד הבא והצג מספר דוגמאות של פלט עבור קלט שונה.

```
n = int(input())
n = abs(n)
count = 1
n //= 10
while n > 0:
    n //= 10
    count += 1
print(count)
```

תרגיל מס' 64.

תפענח את הקוד הבא והצג מספר דוגמאות של פלט עבור קלט שונה.

```
a = input()
a = int(a)
even = 0
odd = 0
while a > 0:
    if a % 2 == 0:
        even += 1
    else:
        odd += 1
    a = a // 10
print("Even: %d, odd: %d" % (even, odd))
```

תרגיל מס' 65.

תפענח את הקוד הבא והצג מספר דוגמאות של פלט עבור קלט שונה.

```
a = input()
even = 0
odd = 0
for i in a:
    if int(i) % 2 == 0:
        even += 1
    else:
        odd += 1
print("Even: %d, odd: %d" % (even, odd))
```



תרגיל מס' 66.

תפענח את הקוד הבא והצג מספר דוגמאות של פלט עבור קלט שונה.

```
a = input()
digits = "02468"
even = 0
odd = 0
for i in a:
    if i in digits:
        even += 1
    else:
        odd += 1
print("Even: %d, odd: %d" % (even, odd))
```

תרגיל מס' 67.

תפענח את הקוד הבא והצג מספר דוגמאות של פלט עבור קלט שונה.

```
n = int(input())
suma = 0
mult = 1
while n > 0:
    digit = n % 10
    suma = suma + digit
    mult = mult * digit
    n = n // 10
print("Sum:", suma)
print("Multiplication:", mult)
```

תרגיל מס' 68.

תפענח את הקוד הבא והצג מספר דוגמאות של פלט עבור קלט שונה.

```
n = input()
suma = 0
mult = 1
for digit in n:
    if digit.isdigit():
        suma += int(digit)
        mult *= int(digit)
print("Sum is:", suma)
print("Multiplication:", mult)
```

תרגיל מס' 69.

תפענח את הקוד הבא והצג מספר דוגמאות של פלט עבור קלט שונה.

```
n1 = int(input("Enter the integer: "))
n2 = 0
while n1 > 0:
    digit = n1 % 10
    n1 = n1 // 10
    n2 = n2 * 10
    n2 = n2 + digit
print("Answer is:", n2)
```

תרגיל מס' 70.

תפענח את הקוד הבא והצג מספר דוגמאות של פלט עבור קלט שונה.

```
i = int(input("The exponent is: "))
n = int(input("The last number is: "))
p = 1
while i ** p <= n:
    print(i ** p, end=' ')
    p += 1
```

תרגיל מס' 71.

תפענח את הקוד הבא והצג מספר דוגמאות של פלט עבור קלט שונה.

```
n = int(input())
factorial = 1
while n > 1:
    factorial *= n
    n -= 1
print(factorial)
```

תרגיל מס' 72.

תפענח את הקוד הבא והצג מספר דוגמאות של פלט עבור קלט שונה.

```
n = int(input())
factorial = 1
for i in range(2, n + 1):
    factorial *= i
print(factorial)
```

תרגיל מס' 73.

תפענח את הקוד הבא והצג מספר דוגמאות של פלט עבור קלט שונה.

```
p = int(input("The exponent is: "))
n = int(input("The last number is: "))
i = 1
while i ** p <= n:
    print(i ** p, end=' ')
    i += 1
print("\n Last exponent is:", i - 1)
```

תרגיל מס' 74.

תפענח את הקוד הבא והצג מספר דוגמאות של פלט עבור קלט שונה.

```
fib1 = 0
fib2 = 1
n = input("Enter the Number of queue : ")
n = int(n)
i = 0
while i < n - 2:
    fib_sum = fib1 + fib2
    fib1 = fib2
    fib2 = fib_sum
    i = i + 1
print(fib2)
```

תרגיל מס' 75.

תפענח את הקוד הבא והצג מספר דוגמאות של פלט עבור קלט שונה.

```
fib1 = 0
fib2 = 1
n = int(input("Enter the Number of queue : ")) - 2
while n > 0:
    fib1, fib2 = fib2, fib1 + fib2
    n -= 1
print(fib2)
```

תרגיל מס' 76.

תפענח את הקוד הבא והצג מספר דוגמאות של פלט עבור קלט שונה.

```
fib1 = 0
fib2 = 1

n = input("Number of the element: ")
n = int(n)
print(fib1, " ", fib2, end=" ")
i = 0
while i < n - 2:
    fib_sum = fib1 + fib2
    fib1 = fib2
    fib2 = fib_sum
    i = i + 1
print(fib2, end=" ")
```

תרגיל מס' 77.

תפענח את הקוד הבא והצג מספר דוגמאות של פלט עבור קלט שונה.

```
fib1 = fib2 = 1
n = int(input("Number of the element: ")) - 2
print(fib1, " ", fib2, end=" ")
while n > 0:
    fib1, fib2 = fib2, fib1 + fib2
    n -= 1
print(fib2, end=" ")
```

תרגיל מס' 78.

תפענח את הקוד הבא והצג מספר דוגמאות של פלט עבור קלט שונה.

```
fib1 = fib2 = 1
n = int(input())
if n < 2:
    quit()
print(fib1, end=' ')
print(fib2, end=' ')
for i in range(2, n):
    fib1, fib2 = fib2, fib1 + fib2
    print(fib2, end=' ')
print()
```

תרגיל מס' 79.

תפענח את הקוד הבא והצג מספר דוגמאות של פלט עבור קלט שונה.

```
for i in range(32, 127):
    print(chr(i), end=' ')
    if (i - 1) % 10 == 0:
        print()
print()
```

תרגיל מס' 80.

תפענח את הקוד הבא והצג מספר דוגמאות של פלט עבור קלט שונה.

```
i = 1
while i < 10:
    j = 1
    while j < 10:
        print("%4d" % (i * j), end="")
        j += 1
    print()
    i += 1
```

תרגיל מס' 81.

תפענח את הקוד הבא והצג מספר דוגמאות של פלט עבור קלט שונה.

```
for i in range(1, 10):
    for j in range(1, 10):
        print("%4d" % (i*j), end="")
    print()
```

תרגיל מס' 82.

תפענח את הקוד הבא והצג מספר דוגמאות של פלט עבור קלט שונה.

```
import random
num=random.randint(1,101)
getnum=0
score=0
while getnum!=num:
    score+=1
    getnum=int(input("Please enter the number"))
    if getnum>num:
        print("Your number is larger than my")
```

```

elif getnum < num:
    print("Your number is smaller than my")
else:
    print("You did it with score " + str(score))

```

תרגיל מס' 83.

תפענח את הקוד הבא והצג מספר דוגמאות של פלט עבור קלט שונה.

```

from random import randint
secret_num = randint(1, 100)
user_num = -1
try_count = 1
while secret_num != user_num:
    print("%d Try number: " % try_count, end="")
    user_num = int(input())
    if user_num < secret_num:
        print("More")
    elif user_num > secret_num:
        print("Less")
    else:
        print("Bingo!")
    try_count += 1

```

תרגיל מס' 84.

תפענח את הקוד הבא והצג מספר דוגמאות של פלט עבור קלט שונה.

```

x1 = float(input("Start point: "))
x2 = float(input("End point: "))
step = float(input("Step: "))
if x1 > x2:
    x1, x2 = x2, x1
print("Function: y = -3x**2 - 4x + 20")
print(" x y")
while x1 <= x2:
    y = -3 * x1 ** 2 - 4 * x1 + 20
    print('%5.2f | %7.2f % (x1, y))
    x1 += step

```

תרגיל מס' 85.

תפענח את הקוד הבא והצג מספר דוגמאות של פלט עבור קלט שונה.

```
a = float(input("a = "))
n = int(input("n = "))
product = a
for i in range(1, n):
    product *= (a + i)
print("Result:", product)
```

תרגיל מס' 86.

תפענח את הקוד הבא והצג מספר דוגמאות של פלט עבור קלט שונה.

```
n = input("Number of elements: ")
n = int(n)
a = 1
i = 0
summa = 0
while i < n:
    summa += a
    a = a / -2
    i += 1
print(summa)
```

תרגיל מס' 87.

תפענח את הקוד הבא והצג מספר דוגמאות של פלט עבור קלט שונה.

```
a = 1
b = 2
sign = 1
n = int(input())
n0=0
sum = 0
while n0<n:
    sum += sign * a / b
    n0 += 1
    a = a + 1
    b = b * 2
    sign = -sign
print(sum)
print(n0)
```

תרגיל מס' 88.

תפענח את הקוד הבא והצג מספר דוגמאות של פלט עבור קלט שונה.

```
n = int(input())
b = ""
while n > 0:
    b = str(n % 2) + b
    n = n // 2
print(b)
```

תרגיל מס' 89.

תפענח את הקוד הבא והצג מספר דוגמאות של פלט עבור קלט שונה.

```
num = int(input())
base = int(input("Base from 2 to 9:"))
if not(2 <= base <= 9):
    quit()
newNum=""
while num > 0:
    newNum=str(num%base)+newNum
    num//=base
print(newNum)
```

תרגיל מס' 90.

תפענח את הקוד הבא והצג מספר דוגמאות של פלט עבור קלט שונה.

```
import math
n = int(input())
if n < 2:
    print("Number had to be more than 1")
    quit()
elif n == 2:
    print("This is primary number")
    quit()
i = 2
limit = int(math.sqrt(n))
while i <= limit:
    if n % i == 0:
        print("This is composite number")
        quit()
    i += 1
print("This is primary number")
```



תרגיל מס' 91.

תפענח את הקוד הבא והצג מספר דוגמאות של פלט עבור קלט שונה.

```
a = int(input())
b = int(input())
while a != 0 and b != 0:
    print("a= ", a, "b= ", b)
    if a > b:
        a = a % b
    else:
        b = b % a
print(a + b)
```

תרגיל מס' 92.

תפענח את הקוד הבא והצג מספר דוגמאות של פלט עבור קלט שונה.

```
a = int(input())
b = int(input())
while a != b:
    print("a= ", a, "b= ", b)
    if a > b:
        a = a - b
    else:
        b = b - a
print(a)
```

תרגיל מס' 93.

תפענח את הקוד הבא והצג מספר דוגמאות של פלט עבור קלט שונה.

```
flag=0
n = int(input())
if n < 2:
    print("Number had to be more than 1")
    quit()
elif n == 2:
    print("This is primary number")
    quit()
i = 2
limit = int(n)
while i < limit:
    if n % i == 0:
        flag=1
        print(i,end=" ")
        i += 1
if(flag==0):
    print("This is primary number")
else:
    print("This is composite number")
```

תרגול בנושא הפיכת קוד מוצרה אחת לצורה אחרת.  
תרגיל מס' 1.

תשנה את הקוד כך שבמקום משפט for יהיה משפט while.

```
import random
for i in range(0,5):
    x,y = [random.randrange(-3, 4) for i in range(0,2)]
    print("\n Point (x,y): ({0},{1})".format(x,y))
    if x == 0 and y == 0:
        print("0. Center")
    elif y == 0:
        print("1. Line OX")
    elif x == 0:
        print("2. Line OY")
    else:
        print("3. Not on the axis")
```

תרגיל מס' 2.

תשנה את הקוד כך שבמקום משפט for יהיה משפט while.

```
import math
for i in range(-5,6):
    h = i/2
    x = math.pi * h
    if x > 0:
        y = 2*math.sin(x)
    else:
        y = 6 - x
    print("h = {0} : x = {1:.4f} : f(x) = {2:.4f}".format(h,x,y))
```

תרגיל מס' 3.

תשנה את הקוד כך שבמקום משפט for יהיה משפט while.

```
for x in range(-4,5):
    if x < -2 or x > 2:
        y = 2*x
    else:
        y = -3 * x
    print("x = {0} : f(x) = {1}".format(x,y))
```

תרגיל מס' 4.

תשנה את הקוד כך שבמקום משפט for יהיה משפט while.

```
for i in range(-2,7):
    x = i / 2
    if x <= 0:
        y = -x
    elif x < 2:
        y = x * x
    else:
        y = 4
    print("x = {0} : f(x) = {1}".format(x,y))
```

תרגיל מס' 5.

תשנה את הקוד כך שבמקום משפט while יהיה משפט for.

```
import math
x = -1
while x < 11:
    x_floor = math.floor(x)
    if x < 0:
        y = 0
    elif x_floor%2 == 0:
        y = 1
    else:
        y = -1
    print("x = {0} : f(x) = {1}".format(x,y))
    x += .5
```

תרגיל מס' 6.

תשנה את הקוד כך שבמקום משפט for יהיה משפט while.

```
import random
N = random.randrange(10)+1
K = random.randrange(-10, 10)
print('K = ', K)
print('N = ', N)
for i in range(0, N):
    print(i+1, " : ", K)
```

תרגיל מס' 7.

תשנה את הקוד כך שבמקום משפט for יהיה משפט while.

```
for i in range(-4,5):
    if i == 0:
        s = "zero "
    elif i > 0:
        s = "positive "
    else:
        s = "negative "
    if i != 0:
        if i%2 == 0:
            s += "even "
        else:
            s += "odd "
    s += "number"
    print(i, " : ",s)
```

תרגיל מס' 8.

תשנה את הקוד כך שבמקום משפט for יהיה משפט while.

```
import random
A = random.randrange(10)
n = random.randrange(10)+1
B = A + n
print('A = ', A)
print('B = ', B)
N = 0
for i in range(A+1,B,1):
    N +=1
    print(i, " : ",N)
print("N = ",N)
```

תרגיל מס' 9.

תשנה את הקוד כך שבמקום משפט for יהיה משפט while.

```
import random
B = random.randrange(2,20)
A = random.randrange(1,B)
print('A = ', A)
print('B = ', B)
N = 0
for i in range(B-1, A, -1):
```

```
print(i,end=' ')
N += 1
print()
print("N = ", N)
```

תרגיל מס' 10.

תשנה את הקוד כך שבמקום משפט for יהיה משפט while.

```
import random
A = random.randrange(10)
n = random.randrange(10)+1
B = A + n
print('A = ', A)
print('B = ', B)
S = 0
for i in range(A,B+1,1):
    S += i
    print(i, " : ", S)
print("Sum = ", S)
```

תרגיל מס' 11.

תשנה את הקוד כך שבמקום משפט for יהיה משפט while.

```
import random
A = random.randrange(10)
n = random.randrange(10)+1
B = A + n
print('A = ', A)
print('B = ', B)
P = 1
for i in range(A,B+1,1):
    P *= i
    print(i, " : ", P)
print("Product = ", P)
```

תרגיל מס' 12.

תשנה את הקוד כך שבמקום משפט for יהיה משפט while.

```
import random
A = random.randrange(10)
n = random.randrange(10)+1
B = A + n
```

```

print('A = ', A)
print('B = ', B)
S = 0
for i in range(A,B+1,1):
    S += i*i
    print(i, " : ",i*i, " : ",S)
print("Sum of squares = ",S)

```

תרגיל מס' 13.

תשנה את הקוד כך שבמקום משפט for יהיה משפט while.

```

import random
N = random.randrange(10)
print('N = ', N)
S = 0.0
for i in range(N):
    S += 1/(i+1)
    print(i+1, " : ",1/(i+1), " : ",S)
print("Sum = ",S)

```

תרגיל מס' 14.

תשנה את הקוד כך שבמקום משפט for יהיה משפט while.

```

import random
N = random.randrange(10)
print('N = ', N)
S = 0.0
for i in range(N,2*N+1):
    x = i**2
    S += x
    print(i, " : ",x, " : ",S)
print("Sum = ",S)

```

תרגיל מס' 15.

תשנה את הקוד כך שבמקום משפט for יהיה משפט while.

```

import random
N = random.randrange(1,10)
print('N = ', N)
P = 1.0
for i in range(1,N+1):
    x = 1 + i*0.1

```

```
P *= x
print(i, " : ",x, " : ",P)
print("Product = ",P)
```

תרגיל מס' 16.

תשנה את הקוד כך שבמקום משפט for יהיה משפט while.

```
import random
N = random.randrange(1,15)
print('N = ', N)
S = 0.0
for i in range(1,N+1):
    x = (1 + i*0.1)*(-1)**(i+1)
    S += x
    print(i, " : ",x, " : ",S)
print("Sum = ",S)
```

תרגיל מס' 17.

תשנה את הקוד כך שבמקום משפט for יהיה משפט while.

```
import random
N = random.randrange(1,15)
print('N = ', N)
S = 0.0
for i in range(1,N+1):
    x = 2 * i - 1
    S += x
    print(i, " : ",x, " : ",S)
print("Sum = ",S)
```

תרגיל מס' 18.

תשנה את הקוד כך שבמקום משפט for יהיה משפט while.

```
import random
N = random.randrange(1,10)
print('N = ', N)
A = random.randrange(-10,10)
print('A = ', A)
P = 1
for i in range(1,N+1):
    P *= A
    print(i, " : ",P)
print("Product = ",P)
```

תרגיל מס' 19.

תשנה את הקוד כך שבמקום משפט for יהיה משפט while.

```
import random
N = random.randrange(1,10)
print('N = ', N)
A = random.randrange(-10,10)
print('A = ', A)
P = 1
for i in range(1,N+1):
    P *= A
    print(A," power ", i," equal",P)
```

תרגיל מס' 20.

תשנה את הקוד כך שבמקום משפט for יהיה משפט while.

```
import random
N = random.randrange(1,10)
print('N = ', N)
A = random.randrange(-10,10)
print('A = ', A)
P = 1.0
S = 1.0
for i in range(1,N+1):
    P *= A
    S += P
    print(i," : ", P," : ", S)
print("Result:",S)
```

תרגיל מס' 21.

תשנה את הקוד כך שבמקום משפט for יהיה משפט while.

```
import random
N = random.randrange(1,10)
print('N = ', N)
A = random.randrange(-10,10)
print('A = ', A)
P = 1.0
S = 1.0
for i in range(1,N+1):
    P *= A * (-1)
    S += P
    print(i," : ", P," : ", S)
print("Result:",S)
```



תרגיל מס' 22.

תשנה את הקוד כך שבמקום משפט for יהיה משפט while.

```
import random
N = random.randrange(1,15)
print('N = ', N)
F = 1.0
for i in range(1,N+1):
    F *= i
    print(i, " : ", F)
print("Result:",F)
```

תרגיל מס' 23.

תשנה את הקוד כך שבמקום משפט for יהיה משפט while.

```
import random
N = random.randrange(1,10)
print('N = ', N)
F = 1.0
S = 0.0
for i in range(1,N+1):
    F *= i
    S += F
    print(i, " : ", F, " : ", S)
print("Result:",S)
```

תרגיל מס' 24.

תשנה את הקוד כך שבמקום משפט for יהיה משפט while.

```
import random
import math
N = random.randrange(1,15)
N = 20
print('N = ', N)
F = 1.0
S = 1.0
for i in range(1,N+1):
    F /= i
    S += F
    print(i, " : ", F, " : ", S)
print("Result:",S)
print("e = ",math.exp(1))
```

תרגיל מס' 25.

תשנה את הקוד כך שבמקום משפט for יהיה משפט while.

```
import math
X = 2
N = 20
print('X = ', X)
print('N = ', N)
F = 1.0
S = 1.0
for i in range(1,N+1):
    F *= X / i
    S += F
    print(i, " : ", F, " : ", S)
print("Result:")
print(S)
print("e:")
print(math.exp(X))
```

תרגיל מס' 26.

תשנה את הקוד כך שבמקום משפט for יהיה משפט while.

```
import math
X = math.pi/4
N = 10
print('X = ', X)
print('N = ', N)
p = X
S = X
k = 0
for i in range(1,N+1):
    k += 2
    p *= (-1) * X * X / (k * (k + 1))
    S += p
    print(i, " : ", k, " : ", k+1, " : ", p, " : ", S)
print("Result:")
print(S)
print("sin:")
print(math.sin(X))
```

תרגיל מס' 27.

תשנה את הקוד כך שבמקום משפט for יהיה משפט while.

```
import math
X = math.pi/3
N = 10
print('X = ', X)
print('N = ', N)
p = 1
S = 1
k = 1
for i in range(1,N+1):
    p *= (-1) * X * X / (k * (k + 1))
    S += p
    print(i, " : ", k, " : ", k+1, " : ", p, " : ", S)
    k += 2
print("Result:")
print(S)
print("cos:")
print(math.cos(X))
```

תרגיל מס' 28.

תשנה את הקוד כך שבמקום משפט for יהיה משפט while.

```
import math
X = 0.5
N = 10
print('X = ', X)
print('N = ', N)
p = 1
S = 0
for i in range(1,N+1):
    p *= X
    S += p/i
    print(i, " : ", p, " : ", S)
    p *= -1
print("Result:")
print(S)
print("ln(x+1):")
print(math.log(X+1))
```

תרגיל מס' 29.

תשנה את הקוד כך שבמקום משפט for יהיה משפט while.

```
import math
X = 0.8
N = 20
print('X = ', X)
print('N = ', N)
p = X
S = X
k = 1
for i in range(1,N+1):
    p *= -1*X*X
    k += 2
    S += p/k
    print(i, " : ", k, " : ", p, " : ", S)
print("Result:")
print(S)
print("atan(x):")
print(math.atan(X))
```

תרגיל מס' 30.

תשנה את הקוד כך שבמקום משפט for יהיה משפט while.

```
import math
X = 0.9
N = 100
print('X = ', X)
print('N = ', N)
p = X
S = X
k = 1
for i in range(1,N+1):
    p *= k/((k+1)*(k+2)) * X*X
    S += p
    print(i, " : ", k, " : ", p, " : ", S)
    k += 2
    p *= k
print("Result:")
print(S)
print("asin(x):")
print(math.asin(X))
```

תרגיל מס' 31.

תשנה את הקוד כך שבמקום משפט for יהיה משפט while.

```
import math
X = -0.9
N = 100
print('X = ', X)
print('N = ', N)
p = X/2
S = 1 + X/2
k = 2
for i in range(2,N+1):
    k += 2
    p *= (-1)*(k-3)/k * X
    S += p
    print(i, " : ", k, " : ", p, " : ", S)
print("Result:")
print(S)
print("sqrt(1+x):")
print(math.sqrt(1+X))
```

תרגיל מס' 32.

תשנה את הקוד כך שבמקום משפט for יהיה משפט while.

```
import random
B = random.randrange(-9,11)
A = random.randrange(-10,B)
N = random.randrange(2,11)
H = (B - A) / N
print("A = ",A)
print("B = ",B)
print("N = ",N)
print("H = ",H)
x = A
for i in range(0,N):
    print(x,end=" ")
    x += H
print(x)
```

תרגיל מס' 33.

תשנה את הקוד כך שבמקום משפט for יהיה משפט while.

```
import random
import math
B = random.randrange(-9,11)
A = random.randrange(-10,B)
N = random.randrange(2,11)
H = (B - A) / N
print("A = ",A)
print("B = ",B)
print("N = ",N)
print("H = {0:.2f}".format(H))
x = A
for i in range(0,N+1):
    y = 1 - math.sin(x)
    print("{0:.2f} : {1:.4f}".format(x,y))
    x += H
```

תרגיל מס' 34.

תשנה את הקוד כך שבמקום משפט for יהיה משפט while.

```
for N in range(1,6,2):
    print("N = ",N)
    A0 = 2
    for k in range(1,N+1):
        A1 = 2 + 1/A0
        print(k, " : ",A1)
        A0 = A1
```

תרגיל מס' 35.

תשנה את הקוד כך שבמקום משפט for יהיה משפט while.

```
for N in range(1,7,2):
    print("N = ",N)
    A0 = 1
    for k in range(1,N+1):
        A1 = (A0 + 1)/k
        print(k, " : ",A1)
        A0 = A1
```

תרגיל מס' 36.

תשנה את הקוד כך שבמקום משפט for יהיה משפט while.

```
for N in range(2,7,2):
    print("N = ",N)
    F1 = 1
    F2 = 1
    print(1, " : ",F1)
    print(2, " : ",F2)
    for k in range(3,N+1):
        F3 = F2 + F1
        print(k, " : ",F3)
        F1 = F2
        F2 = F3
    print()
```

תרגיל מס' 37.

תשנה את הקוד כך שבמקום משפט for יהיה משפט while.

```
for N in range(2,8,2):
    print("N = ",N)
    A1 = 1
    A2 = 2
    print(1, " : ",A1)
    print(2, " : ",A2)
    for k in range(3,N+1):
        A3 = (A1 + 2*A2)/3
        print(k, " : ",A3)
        A1 = A2
        A2 = A3
    print()
```

תרגיל מס' 38.

תשנה את הקוד כך שבמקום משפט for יהיה משפט while.

```
for N in range(2,9,2):
    print("N = ",N)
    A1 = 1
    A2 = 2
    A3 = 3
    print(1, " : ",A1)
    print(2, " : ",A2)
```

```

print(3, " : ",A3)
for k in range(4,N+1):
    A4 = A3 + A2 - 2*A1
    print(k, " : ",A4)
    A1 = A2
    A2 = A3
    A3 = A4
print()

```

תרגיל מס' 39.

תשנה את הקוד כך שבמקום משפט for יהיה משפט while.

```

import random
N = random.randrange(1,11)
K = random.randrange(1,11)
print("N = ",N)
print("K = ",K)
print()
s = 0
for i in range(1,N+1):
    p = 1.0
    for j in range(1,K+1):
        p *= i
    s += p
    print("i = ",i)
    print("p = ",p)
    print("s = ",s)
    print()
print("Sum = ",s)

```

תרגיל מס' 40.

תשנה את הקוד כך שבמקום משפט for יהיה משפט while.

```

import random
N = random.randrange(1,11)
print("N = ",N)
print()
s = 0
for i in range(1,N+1):
    p = 1.0
    for j in range(1,i+1):
        p *= j
    s += p

```



```

print("i = ",i)
print("p = ",p)
print("s = ",s)
print()
print("Sum = ",s)

```

תרגיל מס' 41.

תשנה את הקוד כך שבמקום משפט for יהיה משפט while.

```

import random
N = random.randrange(1,11)
print("N = ",N)
print()
s = 0
for i in range(1,N+1):
    p = 1.0
    K = N + 1 - i
    for j in range(0,K):
        p *= i
    s += p
    print("i = ",i)
    print("j = ",j)
    print("p = ",p)
    print("s = ",s)
    print()
print("Sum = ",s)

```

תרגיל מס' 42.

תשנה את הקוד כך שבמקום משפט for יהיה משפט while.

```

import random
A = random.randrange(1,11)
B = random.randrange(A+1,21)
print("A = ",A)
print("B = ",B)
print()
for i in range(A,B+1):
    for j in range(0,i):
        print(i,end=" ")
    print()

```

תרגיל מס' 43.

תשנה את הקוד כך שבמקום משפט while יהיה משפט for.

```
import random
A = random.randrange(50,100)
B = random.randrange(1,A)
print('A = ', A)
print('B = ', B)
r = A - B
while r >= B:
    r -= B
print("Remainder: ", r)
```

תרגיל מס' 44.

תשנה את הקוד כך שבמקום משפט while יהיה משפט for.

```
import random
A = random.randrange(50,100)
B = random.randrange(1,A)
print('A = ', A)
print('B = ', B)
r = A - B
n = 1
while r >= B:
    r -= B
    n += 1
print(" B in A: ", n)
print("Remainder: ", r)
```

תרגיל מס' 45.

תשנה את הקוד כך שבמקום משפט while יהיה משפט for.

```
N=int(input("Enter the num: "))
print('N = ', N)
while N >= 3:
    N /= 3
print("Really ", (N==1))
```

תרגיל מס' 46.

תשנה את הקוד כך שבמקום משפט while יהיה משפט for.

```
import random
K = random.randrange(1,30)
N = 2 ** K
print('N = ', N)
K_new = 0
while N >= 2:
    N /= 2
    K_new += 1
print("K = ", K_new)
```

תרגיל מס' 47.

תשנה את הקוד כך שבמקום משפט while יהיה משפט for.

```
N = int(input("Enter the number from 3 to 10"))
print('N = ', N)
if N % 2 == 0:
    L = 2
else:
    L = 1
F = 1
while N >= L:
    F *= N
    N -= 2
print("Answer is: ", F)
```

תרגיל מס' 48.

תשנה את הקוד כך שבמקום משפט while יהיה משפט for.

```
import random
N = random.randrange(100, 1000)
print('N = ', N)
K = 1
while K * K <= N:
    K += 1
print("K = {0}, max = {1}, min = {2}".format(K, K ** 2, (K - 1) ** 2))
```

תרגיל מס' 49.

תשנה את הקוד כך שבמקום משפט while יהיה משפט for.

```
import random
N = random.randrange(100,1000)
print('N = ', N)
K = 0
P = 1
while P <= N:
    P *= 3
    K += 1
print("K = {0}, max = {1}, min = {2}".format(K,3**K,3**(K-1)))
```

תרגיל מס' 50.

תשנה את הקוד כך שבמקום משפט while יהיה משפט for.

```
import random
N = random.randrange(5,10)
print('N = ', N)
K = 1
S = 1
while S < N:
    K += 1
    S += K
    print("K = {0}, S = {1}".format(K,S))
print()
print("K = {0}, S = {1}".format(K,S))
```

תרגיל מס' 51.

תשנה את הקוד כך שבמקום משפט while יהיה משפט for.

```
import random
A = random.randrange(2,6)
print('A = ', A)
K = 1.0
S = 1.0
while S <= A:
    K += 1
    x = 1/K
    S += 1/K
    print("K = {0}, 1/K = {1}, S = {2}".format(K,x,S))
print()
print("K = {0}, S = {1}".format(K,S))
```

תרגיל מס' 52.

תשנה את הקוד כך שבמקום משפט for יהיה משפט while .

```
import random
Daily = 10
Total = 200
P = random.randrange(20,51)
print('P = ', P)
coef = 1 + P/100
print("First Day = {0}, Percent = {1}, Coef = {2}".format(Daily,P,coef))
K = 1
S = Daily
while S < Total:
    Daily *= coef
    S += Daily
    K += 1
    print("K = {0}, Daily = {1}, S = {2}".format(K,Daily,S))
print()
print("Days = {0}, Summary = {1}".format(K,S))
```

תרגיל מס' 53.

תשנה את הקוד כך שבמקום משפט while יהיה משפט for .

```
import random
N = random.randrange(1000,100000)
print("N = ",N)
q = N
while q >= 1:
    r = q % 10
    print(r,end="; ")
    q = int(q/10)
```

תרגיל מס' 54.

תשנה את הקוד כך שבמקום משפט while יהיה משפט for .

```
import random
N = random.randrange(100,10000000)
print("N = ",N)
q = N
i = 0
s = 0
while q >= 1:
    i += 1
    r = q % 10
    s += r
    print(i, " - ",r, " - ",s)
    q = int(q/10)
print("Num:",i)
print("Sum:",s)
```

תרגיל מס' 55.

תשנה את הקוד כך שבמקום משפט while יהיה משפט for .

```
import random
N = random.randrange(1,10000)
print("N = ",N)
q = N
s = 0
while q >= 1:
    r = q % 10
    q = int(q/10)
    s = s*10 + r
print("New Num:",s)
```

תרגיל מס' 56.

תשנה את הקוד כך שבמקום משפט while יהיה משפט for .

```
import random
N = random.randrange(1,100000)
print("N = ",N)
q = N
i = 0
flag = False
while q >= 1:
```

```

i += 1
r = q % 10
print(i, " - ", r)
if r == 2:
    flag = True
    break
q = int(q/10)
print(flag)

```

תרגיל מס' 57.

תשנה את הקוד כך שבמקום משפט while יהיה משפט for .

```

N = random.randrange(10,1000)
print("N = ",N)
q = N
i = 0
flag = False
while q >= 1:
    i += 1
    r = q % 10
    print(i, " - ", r)
    if r%2 == 1:
        flag = True
        break
    q = int(q/10)
print(flag)

```

תרגיל מס' 58.

תשנה את הקוד כך שבמקום משפט while יהיה משפט for .

```

import random
A = random.randrange(10,1000)
B = random.randrange(10,1000)
print("A = {0}, B = {1}".format(A,B))
while B > 0:
    A,B = B,A%B
print("Answer is: ",A)

```

תרגיל מס' 59.

תשנה את הקוד כך שבמקום משפט while יהיה משפט for .

```
import random
N = random.randrange(10,100000)
print("N = ",N)
F1 = F2 = 1
print(F1,end="; ")
print(F2,end="; ")
while F2 < N:
    F1, F2 = F2, F1+F2
    print(F2,end="; ")
print()
if F2 == N:
    print("TRUE")
else:
    print("FALSE")
```

תרגיל מס' 60.

תשנה את הקוד כך שבמקום משפט while יהיה משפט for .

```
import random
N = random.randrange(10,100000)
print("N = ",N)
F1 = F2 = 1
print(F1,end="; ")
print(F2,end="; ")
while F2 <= N:
    F1, F2 = F2, F1+F2
    print(F2,end="; ")
print()
print(F2)
```

תרגיל מס' 61.

תשנה את הקוד כך שבמקום משפט while יהיה משפט for .

```
import random
N = random.randrange(1,15)
eps = 1/10**N
print("N = ",N)
print("eps = ",eps)
A1 = 2
print(1,":",A1)
```



```

A2 = 2 + 1/A1
k = 2
print(2,":",A2)
while abs(A2 - A1) >= eps:
    A1 = A2
    A2 = 2 + 1/A2
    k += 1
    print(k,":",A2)
print(k,":",A1,":",A2)

```

תרגיל מס' 62.

תשנה את הקוד כך שבמקום משפט while יהיה משפט for .

```

import random
N = random.randrange(1,15)
eps = 1/10**N
print("N = ",N)
print("eps = ",eps)
A1 = 1
A2 = 2
A3 = (A1 + 2*A2)/3
print(1,":",A1)
print(2,":",A2)
print(3,":",A3)
k = 3
while abs(A3 - A2) >= eps:
    A2, A3 = A3, (A2 + 2*A3)/3
    k += 1
    print(k,":",A3)
print()
print(k,":",A2,":",A3)

```

תרגיל מס' 63.

תפענח את הקוד הבא והצג מספר דוגמאות של פלט עבור קלט שונה.

```

n = int(input())
n = abs(n)
count = 1
n //= 10
while n > 0:
    n //= 10
    count += 1
print(count)

```

תרגיל מס' 64.

תשנה את הקוד כך שבמקום משפט while יהיה משפט for .

```
a = input()
a = int(a)
even = 0
odd = 0
while a > 0:
    if a % 2 == 0:
        even += 1
    else:
        odd += 1
    a = a // 10
print("Even: %d, odd: %d" % (even, odd))
```

תרגיל מס' 65.

תשנה את הקוד כך שבמקום משפט for יהיה משפט while .

```
a = input()
even = 0
odd = 0
for i in a:
    if int(i) % 2 == 0:
        even += 1
    else:
        odd += 1
print("Even: %d, odd: %d" % (even, odd))
```

תרגיל מס' 66.

תשנה את הקוד כך שבמקום משפט for יהיה משפט while .

```
a = input()
digits = "02468"
even = 0
odd = 0
for i in a:
    if i in digits:
        even += 1
    else:
        odd += 1
print("Even: %d, odd: %d" % (even, odd))
```

תרגיל מס' 67.

תשנה את הקוד כך שבמקום משפט while יהיה משפט for .

```
n = int(input())
suma = 0
mult = 1
while n > 0:
    digit = n % 10
    suma = suma + digit
    mult = mult * digit
    n = n // 10
print("Sum:", suma)
print("Multiplication:", mult)
```

תרגיל מס' 68.

תשנה את הקוד כך שבמקום משפט for יהיה משפט while .

```
n = input()
suma = 0
mult = 1
for digit in n:
    if digit.isdigit():
        suma += int(digit)
        mult *= int(digit)
print("Sum is:", suma)
print("Multiplication:", mult)
```

תרגיל מס' 69.

תשנה את הקוד כך שבמקום משפט while יהיה משפט for .

```
n1 = int(input("Enter the integer: "))
n2 = 0
while n1 > 0:
    digit = n1 % 10
    n1 = n1 // 10
    n2 = n2 * 10
    n2 = n2 + digit
print("Answer is:", n2)
```

תרגיל מס' 70.

תשנה את הקוד כך שבמקום משפט while יהיה משפט for .

```
i = int(input("The exponent is: "))
n = int(input("The last number is: "))
p = 1
while i ** p <= n:
    print(i ** p, end=' ')
    p += 1
```

תרגיל מס' 71.

תשנה את הקוד כך שבמקום משפט while יהיה משפט for .

```
n = int(input())
factorial = 1
while n > 1:
    factorial *= n
    n -= 1
print(factorial)
```

תרגיל מס' 72.

תשנה את הקוד כך שבמקום משפט for יהיה משפט while .

```
n = int(input())
factorial = 1
for i in range(2, n + 1):
    factorial *= i
print(factorial)
```

תרגיל מס' 73.

תשנה את הקוד כך שבמקום משפט while יהיה משפט for .

```
fib1 = 0
fib2 = 1
n = input("Enter the Number of queue : ")
n = int(n)
i = 0
while i < n - 2:
    fib_sum = fib1 + fib2
    fib1 = fib2
    fib2 = fib_sum
    i = i + 1
print(fib2)
```

תרגיל מס' 74.

תשנה את הקוד כך שבמקום משפט while יהיה משפט for .

```
fib1 = 0
fib2 = 1
n = int(input("Enter the Number of queue : ")) - 2
while n > 0:
    fib1, fib2 = fib2, fib1 + fib2
    n -= 1
print(fib2)
```

תרגיל מס' 75.

תשנה את הקוד כך שבמקום משפט while יהיה משפט for .

```
fib1 = 0
fib2 = 1
n = input("Number of the element: ")
n = int(n)
print(fib1, " ", fib2, end=" ")
i = 0
while i < n - 2:
    fib_sum = fib1 + fib2
    fib1 = fib2
    fib2 = fib_sum
    i = i + 1
print(fib2, end=" ")
```

תרגיל מס' 76.

תשנה את הקוד כך שבמקום משפט while יהיה משפט for .

```
fib1 = fib2 = 1
n = int(input("Number of the element: ")) - 2
print(fib1, " ", fib2, end=" ")
while n > 0:
    fib1, fib2 = fib2, fib1 + fib2
    n -= 1
print(fib2, end=" ")
```

תרגיל מס' 77.

תשנה את הקוד כך שבמקום משפט for יהיה משפט while .

```
fib1 = fib2 = 1
n = int(input())
if n < 2:
    quit()
print(fib1, end=' ')
print(fib2, end=' ')
for i in range(2, n):
    fib1, fib2 = fib2, fib1 + fib2
    print(fib2, end=' ')
print()
```

תרגיל מס' 78.

תשנה את הקוד כך שבמקום משפט for יהיה משפט while .

```
for i in range(32, 127):
    print(chr(i), end=' ')
    if (i - 1) % 10 == 0:
        print()
print()
```

תרגיל מס' 79.

תשנה את הקוד כך שבמקום משפט while יהיה משפט for .

```
i = 1
while i < 10:
    j = 1
    while j < 10:
        print("%4d" % (i * j), end="")
        j += 1
    print()
    i += 1
```

תרגיל מס' 80.

תשנה את הקוד כך שבמקום משפט while יהיה משפט for .

```
import random
num=random.randint(1,101)
getnum=0
```

```

score=0
while getnum!=num:
    score+=1
    getnum=int(input("Please enter the number"))
    if getnum>num:
        print("Your number is larger than my")
    elif getnum<num:
        print("Your number is smaller than my")
    else:
        print("You did it with score " + str(score))

```

תרגיל מס' 81.

תשנה את הקוד כך שבמקום משפט while יהיה משפט for .

```

x1 = float(input("Start point: "))
x2 = float(input("End point: "))
step = float(input("Step: "))
if x1 > x2:
    x1, x2 = x2, x1
print("Function: y = -3x**2 - 4x + 20")
print(" x y")
while x1 <= x2:
    y = -3 * x1 ** 2 - 4 * x1 + 20
    print('%5.2f | %7.2f' % (x1, y))
    x1 += step

```

תרגיל מס' 82.

תשנה את הקוד כך שבמקום משפט while יהיה משפט for .

```

n = input("Number of elements: ")
n = int(n)
a = 1
i = 0
summa = 0
while i < n:
    summa += a
    a = a / -2
    i += 1
print(summa)

```

תרגיל מס' 83.

תשנה את הקוד כך שבמקום משפט while יהיה משפט for .

```
from random import randint
secret_num = randint(1, 100)
user_num = -1
try_count = 1
while secret_num != user_num:
    print("%d Try number: " % try_count, end="")
    user_num = int(input())
    if user_num < secret_num:
        print("More")
    elif user_num > secret_num:
        print("Less")
    else:
        print("Bingo!")
    try_count += 1
```

תרגיל מס' 84.

תשנה את הקוד כך שבמקום משפט for יהיה משפט while .

```
a = float(input("a = "))
n = int(input("n = "))
product = a
for i in range(1, n):
    product *= (a + i)
print("Result:", product)
```

תרגיל מס' 85.

תשנה את הקוד כך שבמקום משפט for יהיה משפט while .

```
for i in range(1, 10):
    for j in range(1, 10):
        print("%4d" % (i*j), end="")
    print()
```



תרגיל מס' 86.

תשנה את הקוד כך שבמקום משפט while יהיה משפט for .

```
a = 1
b = 2
sign = 1
n = int(input())
n0=0
sum = 0
while n0<n:
    sum += sign * a / b
    n0 += 1
    a = a + 1
    b = b * 2
    sign = -sign
print(sum)
print(n0)
```

תרגיל מס' 87.

תשנה את הקוד כך שבמקום משפט while יהיה משפט for .

```
n = int(input())
b = ""
while n > 0:
    b = str(n % 2) + b
    n = n // 2
print(b)
```

תרגיל מס' 88.

תשנה את הקוד כך שבמקום משפט while יהיה משפט for .

```
num = int(input())
base = int(input("Base from 2 to 9:"))
if not(2<=base <=9):
    quit()
newNum=""
while num > 0:
    newNum=str(num%base)+newNum
    num//=base
print(newNum)
```

תרגיל מס' 89.

תשנה את הקוד כך שבמקום משפט while יהיה משפט for.

```
import math
n = int(input())
if n < 2:
    print("Number had to be more than 1")
    quit()
elif n == 2:
    print("This is primary number")
    quit()
i = 2
limit = int(math.sqrt(n))
while i <= limit:
    if n % i == 0:
        print("This is composite number")
        quit()
    i += 1
print("This is primary number")
```

תרגיל מס' 90.

תשנה את הקוד כך שבמקום משפט while יהיה משפט for.

```
a = int(input())
b = int(input())
while a != 0 and b != 0:
    print("a= ", a, "b= ", b)
    if a > b:
        a = a % b
    else:
        b = b % a
print(a + b)
```

תרגיל מס' 91.

תשנה את הקוד כך שבמקום משפט while יהיה משפט for.

```
a = int(input())
b = int(input())
while a != b:
    print("a= ", a, "b= ", b)
    if a > b:
        a = a - b
    else:
        b = b - a
print(a)
```

תרגיל מס' 92.

תשנה את הקוד כך שבמקום משפט while יהיה משפט for .

```
flag=0
n = int(input())
if n < 2:
    print("Number had to be more than 1")
    quit()
elif n == 2:
    print("This is primary number")
    quit()
i = 2
limit = int(n)
while i < limit:
    if n % i == 0:
        flag=1
        print(i,end=", ")
        i += 1
if(flag==0):
    print("This is primary number")
else:
    print("This is composite number")
```

תרגיל מס' 93.

תשנה את הקוד כך שבמקום משפט while יהיה משפט for .

```
p = int(input("The exponent is: "))
n = int(input("The last number is: "))
i = 1
while i ** p <= n:
    print(i ** p, end=' ')
    i += 1
print("\n Last exponent is:", i - 1)
```

## תרגול בנושא כתיבת קוד.

תרגיל מס' 1.

כתוב תוכנית שמחשבת את מספר הספרות שיש למספר הנקלט מלוח מקשים.

תרגיל מס' 2.

כתוב תוכנית הקולטת מלוח מקשים מספר ורושמת על המסך כמה ספרות זוגיות וכמה ספרות אי-זוגיות יש למספר.

תרגיל מס' 3.

כתוב תוכנית הקולטת מלוח מקשים מספר ורושמת על המסך מהו סכום של ספרות של המספר ומה מכפלה של הספרות.

סכום וכפל של כל הספרות של המספר:

לדוגמה: נתון מספר 253. סכום הספרות שווה ל-  $10 = 2 + 5 + 3$ . מכפלה של הספרות –  $30 = 3 * 5 * 2$ .

תרגיל מס' 4.

כתוב תוכנית שמקבלת מספר כלשהו שלם חיובי ומדפיסה מספר בסדר הפוך. נניח אם מספר הנתון 12345 אז תוכנית מדפיסה 54321.

אלגוריתם:

1. מצא את שארית החלוקה ב-10 מהמספר המקורי. לפיכך, אנו משיגים את הספרה האחרונה שלה.
2. הוסף מספר זה למספר החדש.
3. חלקו בעשר את המספר המקורי. כך, אנו נפטרים מהספרה האחרונה בה.
4. שוב, אנו מוצאים את שארית החלוקה בעשרה מה שנוותר ממספר הראשון. זכור נתון זה.
5. הכפל את המספר השני ב-10. כך אנו מגדילים את מספר ספרות לשניים ומעבירים את הספרה הראשונה לספרות של עשרות.
6. הוסף למספר השני שהשיג את הספרה שנשמרה בעבר מהמספר הראשון.

נחזור על שלבים 3-6 עד שהמספר ההתחלתי יקטן לאפס, כלומר עד שנפטר מכל הספרות שלו.

תרגיל מס' 5.

כתוב תוכנית הקולטת מהמשתמש בסיס שהוא רוצה לעלות בחזקות מ-1 עד  $k$ , ומספר הגדול ביותר שהוא רוצה לקבל. תוכנה מדפיסה את כל המספרים שמתאימים לקריטריונים האלה. לדוגמה: המשתמש מכניס 2 ו-1000 ותוכנית מדפיסה:

2, 4, 8, 16, 32, 64, 128, 256, 512

תרגיל מס' 6.

כתוב תוכנית הקולטת מהמשתמש מספר שלם חיובי ומחשבת את העצרת של המספר. לדוגמה, אם משתמש מכניס 5, תוכנית מחשבת  $120 = 5 * 4 * 3 * 2 * 1$ .

תרגיל מס' 7.

כתוב תוכנית הקולטת מהמשתמש מספר שלם חיובי של סדר האיבר בסדרת פיבונצ'י. תוכנית מדפיסה מספר שעומד במקום הזה. סדרת פיבונצ'י מתחילה ב- 1,0 וכל מספר הבא שווה לסכום של שני מספרים הקודמים. לדוגמה: 0, 1, 1, 2, 3, 5, 8, 13, 21

תרגיל מס' 8.

כתוב תוכנית הקולטת מהמשתמש מספר שלם חיובי של סדר האיבר בסדרת פיבונצ'י. תוכנית מדפיסה את הטור עד המספר שנקלט. סדרת פיבונצ'י מתחילה ב- 1,0 וכל מספר הבא שווה לסכום של שני מספרים הקודמים. לדוגמה: 0, 1, 1, 2, 3, 5, 8, 13, 21

תרגיל מס' 9.

הצג על המסך סימני ASCII החל ממספר 32 עד 126.

תרגיל מס' 10.

הדפס את לוח הכפל

תרגיל מס' 11.

המחשב מגריל מספר אקראי שלם בתחום מ-1 עד 100. המשתמש מנסה למצוא מספר הזה, כאשר המחשב עונה האם מספר שמשתמש נוהצג, קטן או גדול מהמספר שמחשב זרק.

כתוב תוכנית שזורק מספר אקראי ואז שואל את המשתמש על מספר ועונה למשתמש את התוצאה. משחק ממשיך עד שמשתמש מגלה את המספר.

תרגיל מס' 12.

המחשב מגריל מספר אקראי שלם בתחום מ-1 עד 100. המשתמש מנסה למצוא מספר הזה, כאשר המחשב עונה האם מספר שמשתמש נוהצג, קטן או גדול מהמספר שמחשב זרק.

כתוב תוכנית שנותנת אופציה למחשב לשחק נגד אדם במקביל עם משחק של אדם נגד מחשב.

תרגיל מס' 13.

נתונה פונקציה  $y = -3x^2 - 4x + 20$ . משתמש מכניס טווח של x וצעד של התקדמות של x והתוכנית מדפיסה ערכים של y בכל הקטע.

תרגיל מס' 14.

כתוב תוכנית שמקבלת מהמשתמש מספר שלם  $a$  ומספר שלם  $n$ . תוכנית מחשבת  $a+(a+1)+(a+2)+\dots+(a+n-1)$  ומדפיסה את התוצאה של החישוב.

תרגיל מס' 15.

נתון טור  $1, -0.5, 0.25, -0.125, \dots$  כתוב תוכנית שמקבלת מהמשתמש כמות איברים ומחזירה סכום של טור עד האיבר שהמשתמש הקליד. לדוגמה אם  $n=3$ , טור יהיה  $1, -0.5, 0.25$ , וסכום של הטור יהיה  $0.75$ .

תרגיל מס' 16.

נתון טור  $1/2, -2/4, 3/8, -4/16, \dots$  כתוב תוכנית שמקבלת מהמשתמש כמות איברים ומחזירה סכום של טור עד האיבר שהמשתמש הקליד. לדוגמה אם  $n=3$  טור יהיה  $1/2 - 2/4 + 3/8$ .

תרגיל מס' 17.

כתוב תוכנית שבעזרת לולאה מעבירה מספר מבסיס 10 לבסיס 2.

תרגיל מס' 18.

כתוב תוכנית שבעזרת לולאה מעבירה מספר מבסיס 10 לבסיס 16.

תרגיל מס' 19.

כתוב תוכנית שבעזרת לולאה מעבירה מספר מבסיס 2 לבסיס 10.

תרגיל מס' 20.

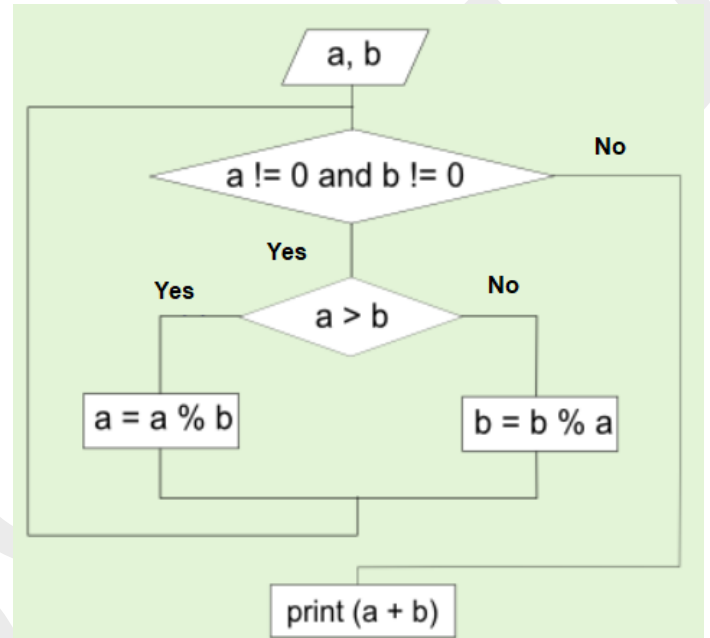
כתוב תוכנית שבעזרת לולאה מעבירה מספר מבסיס 16 לבסיס 10.

תרגיל מס' 21.

כתוב תוכנית הקולטת מהמשתמש מספר ובודקת האם המספר הוא ראשוני או לא. יש להשתמש בשיטת "בדיקת מחלקים".

תרגיל מס' 22.

מצא GCD (מחלק המשותף הגדול ביותר) בעזרת אלגוריתם של אוקלידס.  
אלגוריתם אוקלידס למציאת מכנה משותף הגדול ביותר – GCD.  
אלגוריתם עובד בצורה הבאה:



לדוגמה: צריכים למצוא GCD בין 30 ו-18.

30 לחלק ל-18 = 1 ושארית 12

18 לחלק ל-12 = 1 ושארית 6

12 לחלק ל-6 = 2 ושארית 0

GCD הוא 6.

תרגיל מס' 23.

כתוב תוכנית שמקבלת מהמשתמש מספר בין 2 ל-10, גבול תחתון וגבול עליון ומדפיסה את כל המספרים שמתחלקים במספר הזה ללא שארית.

תרגיל מס' 24.

כתוב תוכנית למימוש פונקציה  $a^x$ .

משתמש מכניס חזקה x ומספר האחרון שמשמש רוצה לראות בטור. תוכנית מעלה את a מ-1 ועד שנגיע לתוצאה שגדולה ממספר המבוקש.

תרגיל מס' 25.

כתוב תוכנית הקולטת מהמשתמש ספרה ומספר איברים ומחשבת סכום איברים לפי דוגמה הבאה:  
נניח ספרה 2 ומספר איברים 5:

$$2 + 22 + 222 + 2222 + 22222 = 24690$$

תרגיל מס' 26.

כתוב תוכנית הקולטת מהמשתמש מספר התחלתי ומספר סופי ומציגה על המסך כל המספרים שנמצאים בין מספרים האלה ומתחלקים ב-3 ללא שארית.

תרגיל מס' 27.

כתוב תוכנית הקולטת מהמשתמש מספר התחלתי ומספר סופי ומציגה על המסך כל המספרים שנמצאים בין מספרים האלה ומתחלקים ב-3 ללא שארית בסדר יורד ומחשב סכומם.

תרגיל מס' 28.

כתוב תוכנית שמקבלת מספר N כלשהו ומחשבת את הסכום:

$$N^2 + (N + 1)^2 + (N + 2)^2 + \dots + (2 \cdot N)^2$$

תרגיל מס' 29.

כתוב תוכנית הקולטת מהמשתמש איבר ראשון של סידרה חשבונית, הפרש וכמות איברים להצגה ומדפיסה את כל האיברים של הסדרה עד האיבר המבוקש.

תרגיל מס' 30.

כתוב תוכנית הקולטת מהמשתמש איבר ראשון של סידרה חשבונית, הפרש והאיבר הגדול ביותר ומדפיסה את כל האיברים של הסדרה עד האיבר המבוקש.

תרגיל מס' 31.

כתוב תוכנית הקולטת מהמשתמש איבר ראשון של סידרה הנדסית, מנה וכמות איברים להצגה ומדפיסה את כל האיברים של הסדרה עד האיבר המבוקש.



תרגיל מס' 32.

כתוב תוכנית הקולטת מהמשתמש איבר ראשון של סידרה הנדסית, מנה והאיבר הגדול ביותר ומדפיסה את כל האיברים של הסדרה עד האיבר המבוקש.

תרגול בנושא לולאות מכוננות – טבלאות מעקב.  
תרגיל מס' 1.

מה יהיה פלט של הקוד הבא:

```
for i in range(5):
    if i == 0 or i == 4:
        for j in range(20):
            print('* ', end="")
    else:
        print('* ', end="")
        for j in range(1, 19):
            print('+ ', end="")
        print('* ', end="")
    print()
```

תרגיל מס' 2.

מה יהיה פלט של הקוד הבא:

```
rows = 5
for i in range(1, rows + 1):
    for j in range(1, i + 1):
        print(j, end=' ')
    print("")
```

תרגיל מס' 3.

מה יהיה פלט של הקוד הבא:

```
rows = 5
b = 0
for i in range(rows, 0, -1):
    b += 1
    for j in range(1, i + 1):
        print(b, end=' ')
    print('\r')
```

תרגיל מס' 4.

מה יהיה פלט של הקוד הבא:

```
rows = 5
num = rows
for i in range(rows, 0, -1):
    for j in range(0, i):
        print(num, end=' ')
    print("\r")
```

תרגיל מס' 5.

מה יהיה פלט של הקוד הבא:

```
rows = 5
for i in range(rows, 0, -1):
    for j in range(0, i + 1):
        print(j, end=' ')
    print("\r")
```

תרגיל מס' 6.

מה יהיה פלט של הקוד הבא:

```
rows = 5
i = 1
while i <= rows:
    j = 1
    while j <= i:
        print((i * 2 - 1), end=" ")
        j += 1
    i += 1
    print("")
```

תרגיל מס' 7.

מה יהיה פלט של הקוד הבא:

```
rows = 5
for i in range(rows, 0, -1):
    num = i
    for j in range(0, i):
        print(num, end=' ')
    print("\r")
```

תרגיל מס' 8.

מה יהיה פלט של הקוד הבא:

```
rows = 6
for i in range(1, rows):
    for j in range(i, 0, -1):
        print(j, end=' ')
    print("")
```

תרגיל מס' 9.

מה יהיה פלט של הקוד הבא:

```
rows = 5
for i in range(0, rows + 1):
    for j in range(rows - i, 0, -1):
        print(j, end=' ')
    print()
```

תרגיל מס' 10.

מה יהיה פלט של הקוד הבא:

```
start = 1
stop = 2
current_num = stop
for row in range(2, 6):
    for col in range(start, stop):
        current_num -= 1
        print(current_num, end=' ')
    print("")
    start = stop
    stop += row
    current_num = stop
```

תרגיל מס' 11.

מה יהיה פלט של הקוד הבא:

```
rows = 6
for i in range(1, rows):
    num = 1
    for j in range(rows, 0, -1):
```

```
if j > i:
    print(" ", end=' ')
else:
    print(num, end=' ')
    num += 1
print("")
```

תרגיל מס' 12.

מה יהיה פלט של הקוד הבא:

```
rows = 5
for i in range(1, rows + 1):
    for j in range(1, rows + 1):
        if j <= i:
            print(i, end=' ')
        else:
            print(j, end=' ')
    print()
```

תרגיל מס' 13.

מה יהיה פלט של הקוד הבא:

```
rows = 8
for i in range(1, rows + 1):
    for j in range(1, i + 1):
        square = i * j
        print(i * j, end=' ')
    print()
```

תרגיל מס' 14.

מה יהיה פלט של הקוד הבא:

```
rows = 5
for i in range(0, rows):
    for j in range(0, i + 1):
        print("*", end=' ')
    print("\r")
```

תרגיל מס' 15.

מה יהיה פלט של הקוד הבא:

```
rows = 5
k = 2 * rows - 2
for i in range(0, rows):
    for j in range(0, k):
        print(end=" ")
    k = k - 2
    for j in range(0, i + 1):
        print("* ", end="")
    print("")
```

תרגיל מס' 16.

מה יהיה פלט של הקוד הבא:

```
rows = 5
for i in range(rows + 1, 0, -1):
    for j in range(0, i - 1):
        print("*", end=' ')
    print(" ")
```

תרגיל מס' 17.

מה יהיה פלט של הקוד הבא:

```
rows = 5
k = 2 * rows - 2
for i in range(rows, -1, -1):
    for j in range(k, 0, -1):
        print(end=" ")
    k = k + 1
    for j in range(0, i + 1):
        print("*", end=" ")
    print("")
```

תרגיל מס' 18.

מה יהיה פלט של הקוד הבא:

```
rows = 5
i = rows
while i >= 1:
```

```

j = rows
while j > i:
    print(' ', end=' ')
    j -= 1
k = 1
while k <= i:
    print('*', end=' ')
    k += 1
print()
i -= 1

```

תרגיל מס' 19.

מה יהיה פלט של הקוד הבא:

```

print("Print equilateral triangle Pyramid using asterisk symbol ")
size = 7
m = (2 * size) - 2
for i in range(0, size):
    for j in range(0, m):
        print(end=" ")
    m = m - 1
    for j in range(0, i + 1):
        print("* ", end=' ')
    print(" ")

```

תרגיל מס' 20.

מה יהיה פלט של הקוד הבא:

```

rows = 6
for i in range(0, rows):
    for j in range(0, i + 1):
        print("*", end=' ')
    print(" ")
for i in range(rows + 1, 0, -1):
    for j in range(0, i - 1):
        print("*", end=' ')
    print(" ")

```

תרגיל מס' 21.

מה יהיה פלט של הקוד הבא:

```
rows = 5
for i in range(0, rows):
    for j in range(0, i + 1):
        print("*", end=' ')
    print("\r")
for i in range(rows, 0, -1):
    for j in range(0, i - 1):
        print("*", end=' ')
    print("\r")
```

תרגיל מס' 22.

מה יהיה פלט של הקוד הבא:

```
rows = 5
i = 1
while i <= rows:
    j = i
    while j < rows:
        print(' ', end=' ')
        j += 1
    k = 1
    while k <= i:
        print('*', end=' ')
        k += 1
    print()
    i += 1
i = rows
while i >= 1:
    j = i
    while j <= rows:
        print(' ', end=' ')
        j += 1
    k = 1
    while k < i:
        print('*', end=' ')
        k += 1
    print("")
    i -= 1
```

תרגיל מס' 23.

מה יהיה פלט של הקוד הבא:

```
rows = 5
i = 0
while i <= rows - 1:
    j = 0
    while j < i:
        print(", end=' ')
        j += 1
    k = i
    while k <= rows - 1:
        print('*', end=' ')
        k += 1
    print()
    i += 1
i = rows - 1
while i >= 0:
    j = 0
    while j < i:
        print(", end=' ')
        j += 1
    k = i
    while k <= rows - 1:
        print('*', end=' ')
        k += 1
    print()
    i -= 1
```

תרגיל מס' 24.

מה יהיה פלט של הקוד הבא:

```
rows = 14
print("*" * rows, end="\n")
i = (rows // 2) - 1
j = 2
while i != 0:
    while j <= (rows - 2):
        print("*" * i, end="")
        print("_" * j, end="")
        print("*" * i, end="\n")
        i = i - 1
        j = j + 2
```



תרגיל מס' 25.

מה יהיה פלט של הקוד הבא:

```
rows = 5
k = 2 * rows - 2
for i in range(0, rows):
    for j in range(0, k):
        print(end=" ")
    k = k - 1
    for j in range(0, i + 1):
        print("* ", end="")
    print("")
k = rows - 2
for i in range(rows, -1, -1):
    for j in range(k, 0, -1):
        print(end=" ")
    k = k + 1
    for j in range(0, i + 1):
        print("* ", end="")
    print("")
```

תרגיל מס' 26.

מה יהיה פלט של הקוד הבא:

```
rows = 5
i = 1
while i <= rows:
    j = rows
    while j > i:
        print(' ', end=' ')
    j -= 1
    print('*', end=' ')
    k = 1
    while k < 2 * (i - 1):
        print(' ', end=' ')
        k += 1
    if i == 1:
        print()
    else:
        print('*')
    i += 1
i = rows - 1
while i >= 1:
    j = rows
```

```

while j > i:
    print(' ', end=' ')
    j -= 1
print('*', end=' ')
k = 1
while k <= 2 * (i - 1):
    print(' ', end=' ')
    k += 1
if i == 1:
    print()
else:
    print('*')
i -= 1

```

תרגיל מס' 27.

מה יהיה פלט של הקוד הבא:

```

ascii_number = 65
rows = 7
for i in range(0, rows):
    for j in range(0, i + 1):
        character = chr(ascii_number)
        print(character, end=' ')
        ascii_number += 1
    print(" ")

```

תרגיל מס' 28.

מה יהיה פלט של הקוד הבא:

```

size = 7
asciiNumber = 65
m = (2 * size) - 2
for i in range(0, size):
    for j in range(0, m):
        print(end=" ")
    m = m - 1
    for j in range(0, i + 1):
        character = chr(asciiNumber)
        print(character, end=' ')
        asciiNumber += 1
    print(" ")

```

תרגיל מס' 29.

מה יהיה פלט של הקוד הבא:

```
character = 'V'  
char_ascii_no = ord(character)  
for i in range(0, 5):  
    for j in range(0, i + 1):  
        char_ascii_no-=1  
        user_char = chr(char_ascii_no)  
        print(user_char, end=' ')  
    print()
```

תרגיל מס' 30.

מה יהיה פלט של הקוד הבא:

```
rows = 10  
for i in range(1, rows + 1):  
    for j in range(1, i + 1):  
        if i*j<10:  
            print(" ",end="")  
        print(i * j, end=' ')  
    print()
```

תרגיל מס' 31.

מה יהיה פלט של הקוד הבא:

```
rows = 9  
for i in range(1, rows):  
    for j in range(-1 + i, -1, -1):  
        print(format(2 ** j, "4d"), end=' ')  
    print("")
```

תרגיל מס' 32.

מה יהיה פלט של הקוד הבא:

```
rows = 9  
for i in range(1, rows):  
    for i in range(0, i, 1):  
        print(format(2 ** i, "4d"), end=' ')  
    for i in range(-1 + i, -1, -1):  
        print(format(2 ** i, "4d"), end=' ')  
    print("")
```

תרגיל מס' 33.

מה יהיה פלט של הקוד הבא:

```
current_num = 1
stop = 2
rows = 4
for i in range(rows):
    for column in range(1, stop):
        print(current_num, end=' ')
        current_num += 1
    print("")
    stop += 2
```

תרגיל מס' 34.

מה יהיה פלט של הקוד הבא:

```
current_num = 1
rows = 4
stop = 2
for i in range(rows):
    for column in range(1, stop):
        print(current_num, end=' ')
        current_num += 1
    print("")
    stop += 1
```

תרגיל מס' 35.

מה יהיה פלט של הקוד הבא:

```
rows = 5
last_num = 2 * rows
even_num = last_num
for i in range(1, rows + 1):
    even_num = last_num
    for j in range(i):
        print(even_num, end=' ')
        even_num -= 2
    print("\r")
```

תרגיל מס' 36.

מה יהיה פלט של הקוד הבא:

```
rows = 6
for i in range(1, rows + 1):
    for j in range(1, i - 1):
        print(j, end=" ")
    for j in range(i - 1, 0, -1):
        print(j, end=" ")
    print()
```

תרגיל מס' 37.

מה יהיה פלט של הקוד הבא:

```
rows = 7
for i in range(0, rows):
    for j in range(0, i + 1):
        print(i * j, end=' ')
    print()
```

תרגיל מס' 38.

מה יהיה פלט של הקוד הבא:

```
rows = 5
for i in range(0, rows + 1, 1):
    for j in range(i + 1, rows + 1, 1):
        print(j, end=' ')
    print()
```

תרגיל מס' 39.

מה יהיה פלט של הקוד הבא:

```
rows = 6
for i in range(0, rows):
    for j in range(rows - 1, i, -1):
        print(j, ", ", end="")
    for l in range(i):
        print(" ", end="")
    for k in range(i + 1, rows):
        print(k, ", ", end="")
    print("\n")
```

תרגיל מס' 40.

מה יהיה פלט של הקוד הבא:

```
row = 4
for i in range(0, row):
    c = 1
    print(c, end=' ')
    for j in range(row - i - 1, 0, -1):
        print('*', end=' ')
        c = c + 1
    print(c, end=' ')
    print("\n")
```

תרגיל מס' 41.

מה יהיה פלט של הקוד הבא:

```
num = 5
counter = 0
for x in range(0, num):
    for y in range(0, x + 1):
        if counter < 10:
            print("", end=" ")
            print(counter, end=" ")
            counter = 2 ** (x + 1)
    print()
```

# תרגול בנושא לולאות מכוננות – כתיבת קוד.

חלק א.

כתוב תוכנית שמדפיסה בעזרת לולאה בתוך לולאה על המסך את הציורים הבאים:

***** ***** ***** ***** *****	***** * * * * * * *****	***** ***** ***** ***** *****	***** ** ** * * * ** ** *****	***** * * * * * * *****	***** ***** ***** ***** *****
***** * * * * ** *	* ** *** **** *****	* ** * * * * *****	* ** * * * * *****	* ** *** **** *****	***** * * * * * * *****
***** ***** ***** *** *	* * * * * * * *****	* *** ***** ***** *****	***** * * * * ** *	***** **** *** ** *	***** ***** **** *** ** *
***** **** *** ** * ** ** *** **** *****	* *** ***** ***** ***** ***** *** *	***** **** ** *** ** ** ** * * * * ** ** *** ** **** ** *****	* ** *** **** ***** **** *** ** *	* ** *** **** ***** **** *** ** *	***** * * * * * * * * *

חלק ב.

כתוב תוכנית שמדפיסה בעזרת לולאה בתוך לולאה על המסך ציורפים הבאים:

11111 11111 11011 11111 11111	11111 10001 10001 10001 11111	01010 01010 01010 01010 01010	11111 00000 11111 00000 11111	12345 21234 32123 43212 54321	10001 01010 00100 01010 10001
12345 23451 34521 45321 54321	55555 54444 54333 54322 54321	12345 23455 34555 45555 55555	11111 22222 33333 44444 55555	12345 23456 34567 45678 56789	10101 01010 10101 01010 10101
12345 1234 123 12 1	5 44 333 2222 11111	11111 2222 333 44 5	55555 4444 333 22 1	1 22 333 4444 55555	12345 21234 32123 43212 54321
5 45 345 2345 12345	54321 5432 543 54 5	5 54 543 5432 54321	54321 4321 321 21 1	1 21 321 4321 54321	1 12 123 1234 12345
1 00 111 0000 11111	1 10 101 1010 10101	13579 3579 579 79 9	56789 4567 345 23 1	1 23 345 4567 56789	12345 2345 345 45 5
1 32 4543 567654 67898765	1 121 12321 1234321 123454321	1 22 333 2222 11111	1 21 123 4321 12345	1 24 135 2468 13579	1 123 12345 1234567 123456789



כתוב תוכנית שמדפיסה בעזרת לולאה בתוך לולאה על המסך צירופים הבאים:

1 23 4567 89123456 7891234567891234	1 1 12 21 123 321 1234 4321 1234554321	1 2 3 4 5 6 7 8 9 10 11 12 13 14 15	1 3 2 4 5 6 10 9 8 7 11 12 13 14 15
1 2 4 7 11 16 22 29 37 46 56 67 79 92 106	1 2 6 3 7 10 4 8 11 13 5 9 12 14 15	1 131 13531 1357531 135797531	2 242 24642 2468642 2468108642
1 12 123 1234 12345 1234 123 12 1	1 2 3 4 5 16 17 18 19 6 15 24 25 20 7 14 23 22 21 8 13 12 11 10 9	1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25	5 5 5 5 5 5 5 5 5 4 4 4 4 4 4 5 5 4 3 3 3 3 4 5 5 4 3 2 2 2 3 4 5 5 4 3 2 1 2 3 4 5 5 4 3 2 2 2 3 4 5 5 4 3 3 3 3 3 4 5 5 4 4 4 4 4 4 5 5 5 5 5 5 5 5 5
	* *1* *121* *12321* *1234321* *123454321* *1234321* *12321* *121* *1* *	1 121 12321 1234321 123454321 1234321 12321 121 1	1 123 12345 1234567 123456789 1234567 12345 123 1

## פרק ה'. פונקציות.

פונקציה בתכנות היא פיסת קוד נפרדת שניתן לקרוא אותה על ידי קריאה אליה בשמה, שלפיו נקרא שמו. כאשר היא "מוזמנת", גוף הפונקציה מבוצע.

ניתן להשוות פונקציות עם תוכנות קטנות שהן לא מבוצעות באופן אוטונומי, אלא מובנות בתוכנה רגילה. לעתים קרובות הן נקראות כך - תת-שגרות. אין הבדלי מפתח אחרים בין פונקציות לתוכניות. פונקציות יכולות גם לקבל ולהחזיר נתונים במידת הצורך. רק בדרך כלל מקבלים אותם לא מהמקלדת, הקובץ וממקורות אחרים, אלא מהתוכנית המתקשרת. הם מחזירים את תוצאת עבודתם לאותו מקום שממנו הזמינו אותם.

יש פונקציות רבות המובנות בשפת התכנות. כבר נתקלנו בכמה כאלה בפייתון: `input`, `print`. הקוד של גופם אינו נראה לעינינו, הוא נמצא במקום כלשהו "מוסתר בתוך השפה". אנו מספקים רק ממשק שהוא - שם הפונקציה.

מצד שני, מתכנת יכול תמיד להגדיר את הפונקציות שלו. הם נקראים מנהג. במקרה זה, "המשתמש" מובן כמתכנת, ולא כמשתמש בתוכנית. נבין מדוע אנו זקוקים לפונקציות אלה, וכיצד ליצור אוהצג. נניח שאנו צריכים 3 פעמים לחבר שני נתונים. ניתן להשתמש בלולאה רגילה מסוג `while` או מסוג `for`.

```
i = 0
while i < 3:
    a = int(input())
    b = int(input())
    print(a+b)
    i += 1
```

```
2
3
5
4
5
9
7
1
8
```

עם זאת, מה אם לפני כל בקשה למספרים, יש צורך להציג כתובת, מדוע הם נדרשים ובכל פעם כתובת זו שונה. איננו יכולים להפריע למחזור ואז לחזור לאותו המעגל בחזרה. יש לנטוש אותו ואז אתה מקבל קוד ארוך המכיל את אותם סעיפים במקומות שונים.

```
print(" How much cats and dogs?")
a = int(input())
b = int(input())
```

```
print("There are ", a+b, "animals")
print("How much lions and tigers?")
a = int(input())
b = int(input())
print("There are ", a+b, "animals")
print("How much snakes and turtles?")
a = int(input())
b = int(input())
print("There are ", a+b, "animals")
```

```
How much cats and dogs?
3
5
There are 8 animals
How much lions and tigers?
8
1
There are 9 animals
How much snakes and turtles?
5
2
There are 7 animals
```

במקרה כזה עדיף להשתמש בפונקציות. הצגת פונקציות מאפשרת לנו לפתור את בעיית שכפול הקוד במקומות שונים של התוכנית. בזכותם ניתן לבצע את אותו קטע קוד לא באופן מיד, אלא רק בעת הצורך.

## הגדרת הפונקציות .def

בשפת פייתון הגדרת הפונקציה נעשה בעזרת אופרטור .def.

```
def countFood():
    a = int(input())
    b = int(input())
    print("There are ", a + b, "animals")
```

זו דוגמה להגדרת פונקציה. כמו הוראות מורכבות אחרות כמו הצהרה מותנית ולולאות, פונקציה מורכבת מכותרת ראשית וגוף. הכותרת מסתיימת בסימן ":". שימו לב על רווחים בשורות של גוף הפונקציה. מילת המפתח המגדירה אומרת למתורגמן שלפניו יש הגדרת הפונקציה. אחרי ההגדרה מופיע שם הפונקציה. שם יכול להיות כל שם שבא לכם, אך כללים הם אותם כללים כמו בהגדרת משתנה. למשל: my\_func1. בתכנות רצוי מאוד לתת שמות משמעותיים לכל דבר. אז במקרה זה, הפונקציה נקראת "num\_of\_animals".

סוגריים ממוקמים על שם הפונקציה. בדוגמה, הם ריקים. משמעות הדבר היא, שהפונקציה אינה מקבלת נתונים מהתוכנית הקוראת לה. עם זאת היא יכלה לקבל אותם, ואז הפרמטרים יהיו בתוך

הסוגריים. אחרי נקודתיים מגיע גוף המכיל הוראות שמבוצעות בעת קריאת הפונקציה. יש להבחין בהגדרת הפונקציה ובקריאתה. בקוד התוכנית הם לא נמצאים בקרבת מקום ולא ביחד. ניתן להגדיר פונקציה, אך לעולם לא לקרוא לה. אי אפשר לקרוא לפונקציה שלא הוגדרה. אם הגדרת פונקציה, אך לעולם לא קראת לה, לעולם לא תבצע את גופה.

```
def countFood():
    a = int(input())
    b = int(input())
    print("There are ", a + b, "animals")

print(" How much cats and dogs?")
countFood()
print("How much lions and tigers?")
countFood()
print("How much snakes and turtles?")
countFood()
```

```
How much cats and dogs?
2
3
There are 5 animals
How much lions and tigers?
1
2
There are 3 animals
How much snakes and turtles?
3
2
There are 5 animals
```

לאחר שכל הודעה מוצגת על המסך, מתבצעת הפעלת הפונקציה שנראית ממש כמו הגדרת הפונקציה יחד עם הסוגריים. מכיוון שאיננו מעבירים שום דבר לפונקציה, הסוגריים שוב ריקים. בקוד לעיל הפונקציה הזאת מופעלת שלוש פעמים.

כאשר הפונקציה מתחילה להתבצע, זרימת ביצוע התוכנית ממשיכה לשורות הגדרת התוכנית והגוף שלה מתחיל להתבצע. לאחר ביצוע גוף הפונקציה, מתבצעת חזרה לשורות הקוד הראשי למקום בו נקראה הפונקציה. בשלב הבא מבוצע הביטוי הבא בקוד.

ב-Python, הגדרת פונקציה חייבת להקדים את הפעלה שלה. זה נובע מהעובדה שהמתורגמן קורא את הקוד שורה אחר שורה ושלא ידוע לו מה נמצא לאחר פקודה שהוא מבצע עכשיו. לפיכך, אם הפעלת הפונקציה קודמת להגדרתה, מתרחשת שגיאה ונזרקת חריגה של `NameError`.

```
print(" How much cats and dogs?")
countFood()
print("How much lions and tigers?")
countFood()
```

```
print("How much snakes and turtles?")
countFood()

def countFood():
    a = int(input())
    b = int(input())
    print("There are ", a + b, "animals")
```

```
Traceback (most recent call last):
  File "C:/MyPython/third.py", line 3, in <module>
    countFood()
NameError: name 'countFood' is not defined
How much cats and dogs?
```

## פונקציות הופכות תוכנה למובנת ומסודרת.

היתרון הנוסף של הפונקציות - זה לא רק היכולת לקרוא שוב ושוב לאותו קוד ממקומות שונים בתוכנית- לא פחות חשוב שבזכותם התוכנית נראת מסודרת ומובנת יותר.

נראה שהפונקציות מחלקות את התוכנית לחלקים נפרדים, שכל אחד מהם מבצע את המשימה הספציפית שלו.

נכתוב תוכנית המחשבת את שטח הצורות השונות. המשתמש מציין את סוג הצורה שהוא רוצה לחשב את השטח שלו. לאחר מכן הוא מזין את הצלעות.

ל דוגמה, האורך והרוחב במקרה של מלבן או רדיוס של עיגול. כדי לפצל ביצוע למספר ענפים, נשתמש באופרטור if-elif-else:

## נכתוב תוכנית פעם ללא פונקציות ופעם עם פונקציות:

```
figure = input("1-rectangle, 2-triangle, 3-circle: ")
if figure == '1':
    a = float(input("Length: "))
    b = float(input("Width: "))
    print("Square is: %.2f" % (a*b))
elif figure == '2':
    a = float(input("Base: "))
    h = float(input("Height: "))
    print("Square is: %.2f" % (0.5 * a * h))
elif figure == '3':
    r = float(input("Radius: "))
    print("Square is: %.2f" % (3.14 * r**2))
else:
    print("Error of input")
```

```
1-rectangle, 2-triangle, 3-circle: 3
Radius: 2
Square is: 12.56
```

```
def rectangle():
    a = float(input("Length: "))
    b = float(input("Width: "))
    print("Square is: %.2f" % (a * b))

def triangle():
    a = float(input("Base: "))
    h = float(input("Height: "))
    print("Square is: %.2f" % (0.5 * a * h))

def circle():
    r = float(input("Radius: "))
    print("Square is: %.2f" % (3.14 * r ** 2))

figure = input("1-rectangle, 2-triangle, 3-circle: ")
if figure == '1':
    rectangle()
elif figure == '2':
    triangle()
elif figure == '3':
    circle()
else:
    print("Error of input")
```

```
1-rectangle, 2-triangle, 3-circle: 1
Length: 4
Width: 6
Square is: 24.00
```

זה נראה מסובך יותר, וכל אחת משלושת הפונקציות מופעלות רק פעם אחת. עם זאת, מההיגיון הכללי של התוכנית, הוראות לאיתור אזורים הוסרו ובודדו.

התוכנית מורכבת כעת מ"אבני לגו" נפרדות. בתוכנה הראשית, נוכל לשלב אותם כרצונכם. היא משחקת את התפקיד של מנגנון בקרה. אם אי פעם נרצה לחשב את שטח המשולש באמצעות הנוסחה אחרת ולא דרך הגובה, לא נצטרך לחפש את הקוד בתוכנה כולה, דמיון שהוא מורכב מאלפי שורות קוד כמו תוכניות אמיתיות. בשימוש בפונקציה מוגדרת מראש, כל שעלינו לעשות הוא לשנות את הקוד בגוף הפונקציה כדי להשפיע על כל המקומות בתוכנית שבהם הפונקציה מופעלת.

אם צריך להשתמש בפונקציות אלה בכל תוכנית אחרת, נוכל לייבא אותם לשם, בהתייחס לקובץ זה עם הקוד כפי שהוא נעשה בנייתו, נראה בהמשך.

[האם סדר של הגדרות של פונקציות חשוב?](#)  
נסדר אותם בסדר אחר ונפעיל עוד פעם את התוכנית.

```
def triangle():
    a = float(input("Base: "))
    h = float(input("Height: "))
    print("Square is: %.2f" % (0.5 * a * h))

def circle():
    r = float(input("Radius: "))
    print("Square is: %.2f" % (3.14 * r ** 2))

def rectangle():
    a = float(input("Length: "))
    b = float(input("Width: "))
    print("Square is: %.2f" % (a * b))

figure = input("1-rectangle, 2-triangle, 3-circle: ")
if figure == '1':
    rectangle()
elif figure == '2':
    triangle()
elif figure == '3':
    circle()
else:
    print("Error of input")
```

```
1-rectangle, 2-triangle, 3-circle: 2
Base: 2
Height: 4
Square is: 4.00
```

מושג `*args` ומושג `**kwargs`.

התחביר המיוחד `*args` בהגדרות פונקציה בפייתון משמש להעברת מספר משתנים של ארגומנטים לפונקציה. הוא משמש להעברת רשימת טיעונים שאינם מילות מפתח, באורך משתנה.

התחביר הוא להשתמש בסמל `*` כדי לקחת מספר משתנה של טיעונים על פי המוסכמה, משתמשים בה לעתים קרובות במילה `args`.

מה שהמטוען `*` מאפשר לך לעשות הוא להכניס יותר טיעונים ממספר הטיעונים הרשמיים שהגדרת בעבר. באמצעות `*args`, ניתן להתאים לכל מספר של טיעונים נוספים לפרמטרים הרשמיים הנוכחיים שלך (כולל אפס טיעונים נוספים).

לדוגמה: אנו רוצים לבצע פונקציית כפל שלוקחת מספר ארגומנטים כלשהו ומסוגלת להכפיל את כולם יחד. ניתן לעשות זאת באמצעות \*args.

השימוש ב- \*, המשתנה שאנו מקשרים אליו \* הופך למשמעות חוזרת ואפשר לעשות דברים כמו איטרציה מעליו, להפעיל כמה פונקציות בסדר גודל גבוה יותר כמו מפה, מסנן וכו'.

```
# Python program to illustrate
# *args for variable number of arguments
def myFun(*argv):
    for arg in argv:
        print(arg,end="")
myFun('Hello ', 'Welcome ', 'to ', 'www.','elecstudy','.org')
```

```
C:\Users\marka\AppData\Local\Programs\Python\Python38-32\python.exe C:/MyPython/third.py
Hello Welcome to www.elecstudy.org
```

דוגמה אחרת:

```
# Python program to illustrate
# *args with first extra argument
def myFun(arg1, *argv):
    print("First argument :", arg1)
    for arg in argv:
        print("Next argument through *argv :", arg)
myFun('You', 'are', 'studying', 'Python')
```

```
First argument : You
Next argument through *argv : are
Next argument through *argv : studying
Next argument through *argv : Python
```

התחביר המיוחד \*\* kwargs בהגדרות פונקציה בפייתון משמש להעברת רשימת טיעונים באורך משתנה עם מילות מפתח. אנו משתמשים בשם kwargs עם הכוכב הכפול. הסיבה היא מכיוון שהכוכב הכפול מאפשר לנו לעבור טיעוני מילות מפתח (וכל מספר מהם). טיעון מילות מפתח הוא המקום בו אתה מספק שם למשתנה כשאתה מעביר אותו לפונקציה. אפשר לחשוב על kwargs כמילון שממפה כל מילת מפתח לערך שאנחנו מעבירים לצידו. זו הסיבה שכאשר אנו חוזרים על kwargs לא נראה שיש סדר שהם הודפסו.



```
# Python program to illustrate
# *kargs for variable number of keyword arguments

def myFun(**kwargs):
    for key, value in kwargs.items():
        print("%s == %s" % (key, value))

myFun(first='We', mid='are', last='the',final='champions')
```

```
first == We
mid == are
last == the
final == champions
```

דוגמה אחרת:

```
# Python program to illustrate **kargs for
# variable number of keyword arguments with
# one extra argument.

def myFun(arg1, **kwargs):
    for key, value in kwargs.items():
        print("%s == %s" % (key, value))

myFun("Hi", first='First', mid='Second', last='Third')
```

```
first == First
mid == Second
last == Third
```

שימוש ב- **\*args** ו- **\*\*kwargs** לקריאת פונקציה.

```
def myFun(arg1, arg2, arg3):
    print("arg1:", arg1)
    print("arg2:", arg2)
    print("arg3:", arg3)

# Now we can use *args or **kwargs to
# pass arguments to this function :
args = ("Electronics", "Electricity", "Programming")
myFun(*args)
```

```
kwargs = {"arg1": "Electronics", "arg2": "Electricity", "arg3": "Program"}
myFun(**kwargs)
```

```
arg1: Electronics
arg2: Electricity
arg3: Program
arg1: Electronics
arg2: Electricity
arg3: Program
```

שימוש ב- `*args` ו- `**kwargs` לקריאת פונקציה באותה שורה.

```
def myFun(*args, **kwargs):
    print("args: ", args)
    print("kwargs: ", kwargs)

# Now we can use both *args, **kwargs to pass arguments to this function :
myFun('first', 'second', 'third', first="Electronics", mid="Electricity", last="Programming")
```

```
args: ('first', 'second', 'third')
kwargs: {'first': 'Electronics', 'mid': 'Electricity', 'last': 'Programming'}
```

## משתנים לוקליים וגלובליים:

בתכנות מוקדשת תשומת לב מיוחדת למושג המשתנים המקומיים (לוקליים) והעולמיים (גלובליים), כמו גם למושג "אזור נראות". בהתאם לכך, משתנים מקומיים גלויים רק בהיקף המקומי שיכול להיות פונקציה יחידה. משתנים גלובליים גלויים לאורך התוכנית. "גלוי" פירושו ידוע, נגיש. ניתן לגשת אליהם לפי שם ולקבל את הערך המשוך אליהם. ניתן לגשת למשתנה גלובלי מהיקף מקומי. לא ניתן לגשת למשתנה מקומי מההיקף הגלובלי מכיוון שהמשתנה המקומי קיים רק בגוף הפונקציה הבודדת. כשיוצאים מהפונקציה, משתנים מקומיים נעלמים (נמחקים מהזיכרון). משוחרר זיכרון המחשב שהוקצה להם. כאשר נקראת הפונקציה שוב, ייווצרו משתנים מקומיים שוב. (פעם אזור הזיכרון של נתונים היה בסך הכול 65 קילו בית והיה חשוב מאוד לשחרר את הזיכרון כדי שיהיה מספיק מקום לכל הנתונים).

נראה את הדוגמה משיעור הקודם ונבדוק איזה משתנים הם משתנים לוקליים ואיזה גלובליים.

```
def triangle():
    a = float(input("Base: "))
    h = float(input("Height: "))
    print("Square is: %.2f" % (0.5 * a * h))

def circle():
    r = float(input("Radius: "))
    print("Square is: %.2f" % (3.14 * r ** 2))
```

```

def rectangle():
    a = float(input("Length: "))
    b = float(input("Width: "))
    print("Square is: %.2f" % (a * b))

figure = input("1-rectangle, 2-triangle, 3-circle: ")
if figure == '1':
    rectangle()
elif figure == '2':
    triangle()
elif figure == '3':
    circle()
else:
    print("Error of input")

```

יש כאן מספר משתנים: משתנה figure, הוא משתנה היחיד שהוא גלובלי – הוא מוגדר בתוכנה ראשית וניתן להשתמש בו בכל קטע של תוכנית. לעומת זאת משתנים a,h,r,b הם משתנים לוקליים. נניח ונרצה להדפיס בתוכנה ראשית את רדיוס של המעגל:

```

def triangle():
    a = float(input("Base: "))
    h = float(input("Height: "))
    print("Square is: %.2f" % (0.5 * a * h))

def circle():
    r = float(input("Radius: "))
    print("Square is: %.2f" % (3.14 * r ** 2))

def rectangle():
    a = float(input("Length: "))
    b = float(input("Width: "))
    print("Square is: %.2f" % (a * b))

figure = input("1-rectangle, 2-triangle, 3-circle: ")
if figure == '1':
    rectangle()
elif figure == '2':
    triangle()
elif figure == '3':
    circle()
    print(r)
else:
    print("Error of input")

```

בזמן הקומפילציה לא נקבל שום שגיאה, אבל בזמן ההרצה נקבל שגיאת ההרצה.

```
1-rectangle, 2-triangle, 3-circle: 3
Radius: 12
Square is: 452.16
Traceback (most recent call last):
  File "C:/MyPython/third.py", line 24, in <module>
    print(r)
NameError: name 'r' is not defined
```

בהבדל משפות אחרות (בשפת C במקרה כזה נקבל הודעת שגיאה בזמן הקומפילציה) לא נקבל הודעת שגיאה, אם לא נזמין את הפונקציה הבעייתית.

```
1-rectangle, 2-triangle, 3-circle: 1
Length: 2
Width: 4
Square is: 8.00
```

שמות של פונקציות `rectangle()`, `triangle()`, `circle()` הם גם שמות שיש להם אזור קיום. כאן במקרה הן פונקציות גלובליות, אבל יכול להיות מצב שנגדיר פונקציות לוקליות (כדי לשחרר זיכרון) ואז בחלק מהאזור הקוד של הפונקציות לא יתקיים כלל.

## אין בעיה אם אנו נפנה למשתנים גלובליים מפונקציה ראשית.

```
def triangle():
    a = float(input("Base %s: " %figure))
    h = float(input("Height %s: " %figure))
    print("Square is: %.2f" % (0.5 * a * h))

def circle():
    r = float(input("Radius %s: " %figure))
    print("Square is: %.2f" % (3.14 * r ** 2))

def rectangle():
    a = float(input("Length %s: " %figure))
    b = float(input("Width %s: " %figure))
    print("Square is: %.2f" % (a * b))

figure = input("1-rectangle, 2-triangle, 3-circle: ")
if figure == '1':
    figure=" rectangle "
    rectangle()
elif figure == '2':
    figure = " triangle "
    triangle()
elif figure == '3':
    figure = " circle "
    circle()
else:
    print("Error of input")
```

```
1-rectangle, 2-triangle, 3-circle: 3
Radius circle : 5
Square is: 78.50
```

במקרה זה, הפנייה למשתנה בשם figure אפשרית, עקב העובדה שהוא הוכרז בהיקף הגלובלי ונראה בתוכנית כולה.

הפונקציות שלנו לא ממש מושלמות. עליהן לחשב את שטח הצורות, אך הן לא צריכות להציג את התוצאה על המסך. סביר להניח שהתוצאה נדרשת לצרכים הפנימיים של התוכנית, לחישובים נוספים ולהציג את תוצאת החישובים הנוספים על המסך.

אם הפונקציות לא יציגו אלא רק יחשבו את התוצאה, עליהן להיות מאוחסנות במקום כלשהו לשימוש עתידי. משתנים גלובליים יתאימו לכך, אליהם ניתן לכתוב את התוצאה.

נכתוב את התוכנית כך:

```
def triangle():
    a = float(input("Base %s: " %figure))
    h = float(input("Height %s: " %figure))
    result=0.5 * a * h

def circle():
    r = float(input("Radius %s: " %figure))
    result=3.14 * r ** 2

def rectangle():
    a = float(input("Length %s: " %figure))
    b = float(input("Width %s: " %figure))
    result=a * b

result=0
figure = input("1-rectangle, 2-triangle, 3-circle: ")

if figure == '1':
    figure=" rectangle "
    rectangle()
elif figure == '2':
    figure = " triangle "
    triangle()
elif figure == '3':
    figure = " circle "
    circle()
else:
    print("Error of input")
print("Square is: %.2f" % result)
```

מתברר שפונקציה עובדת לא כמו שציפינו:

```
1-rectangle, 2-triangle, 3-circle: 1
Length rectangle : 2
Width rectangle : 4
Square is: 0.00
```

העניין הוא שבפייתון, הקצאת ערך למשתנה משולבת בהכרחו. בשפות רבות אחרות זה לא כך. לפיכך, כאשר השם result מוזכר לראשונה בהיקף המקומי, ובמקביל הערך מוקצה לו, נוצר משתנה מקומי result. זהו משתנה נוסף שלא קשור לתוצאות גלובליות.

עם סיום הפונקציה, ערך התוצאה המקומי הולך לאיבוד והערך הגלובלי לא השתנה.

כאשר קראנו למשתנה figure בתוך הפונקציה, לא הקצנו לה כלום.

להפך, ביקשנו את משמעותה. המתורגמן של פייתון חיפש תחילה את משמעותו בהיקף המקומי ולא מצא אותו, ואחרי זה הלך להיקף הגלובלי ומצא אותו.

במקרה של תוצאות הוא לא מחפש דבר. הוא מבצע חישובים מימין לסימן ההקצאה, יוצר תוצאה, ומקצה את ערך התוצאה למשתנה מקומי.

למעשה, ניתן לכפות גישה למשתנה גלובלי. יש לזה פקודה הנקראת global.

```
def triangle():
    a = float(input("Base %s: " %figure))
    h = float(input("Height %s: " %figure))
    global result
    result=0.5 * a * h

def circle():
    r = float(input("Radius %s: " %figure))
    global result
    result=3.14 * r ** 2

def rectangle():
    a = float(input("Length %s: " %figure))
    b = float(input("Width %s: " %figure))
    global result
    result=a * b

result=0
figure = input("1-rectangle, 2-triangle, 3-circle: ")

if figure == '1':
    figure=" rectangle "
    rectangle()
elif figure == '2':
    figure = " triangle "
    triangle()
elif figure == '3':
    figure = " circle "
    circle()
else:
    print("Error of input")
print("Square is: %.2f" % result)
```

1-rectangle, 2-triangle, 3-circle: 1  
Length rectangle : 2  
Width rectangle : 4  
Square is: 8.00

בגרסה זו התוכנית תעבוד נכון.

עם זאת, שינוי ערכי המשתנים הגלובליים בגוף הפונקציה הוא תרגול רע. בתוכניות גדולות קשה למתכנת לעקוב אחר היכן, איזו פונקציה ולמה המשמעות שלהם השתנתה. המתכנת בוחן את הערך ההתחלתי של המשתנה הגלובלי ויכול לחשוב שהוא נשאר זהה. קשה לשים לב שפונקציה מסוימת שינתה אותה.

זה מוביל לטעויות לוגיות.

כדי להפטר מהצורך להשתמש במשתנים גלובליים, פונקציות אפשריות להחזיר את תוצאת עבודתם לענף הראשי של התוכנית. וכבר ניתן להקצות ערך זה המתקבל מהפונקציה למשתנה גלובלי בהיקף הגלובלי. זה הופך את התוכנית למובנת יותר.

## החזרת פרמטרים מהפונקציות.

פונקציות יכולות להעביר כל מידע מגופם לענף הראשי של התוכנית.

כלומר, הפונקציה מחזירה ערך. ברוב שפות התכנות, כולל פייתון, יציאה מהפונקציה והעברת נתונים למקום ממנו נקראת, מבוצעת על ידי אופרטור return.

אם המתורגמן של פייתון, בעת ביצוע גוף הפונקציה, נתקל ב- return, אז הוא "מרים" את הערך שצוין לאחר פקודה זו ו- "משחרר" את הפונקציה.

```
def cylinder():  
    r = float(input("Radius is"))  
    h = float(input("Height is"))  
    # square of the side of cylinder  
    side = 2 * 3.14 * r * h  
    # square of the base  
    circle = 3.14 * r**2  
    # total square of cylinder  
    full = side + 2 * circle  
    return full  
  
square = cylinder()  
print(square)
```

Radius is 2  
Height is 1  
37.68



בתוכנית זו, הערך של המשתנה המקומי full מוחזר לענף הראשי מהפונקציה. לא המשתנה עצמו, אלא ערכו, במקרה זה, מספר כלשהו המתקבל בחישוב שטח הצילינדר.

בענף הראשי של התוכנית, ערך זה מוקצה למשתנה הגלובלי square.

כלומר, הביטוי `square = cylinder()` מבוצע כך:

1. קוראים את הפונקציה `cylinder()`

2. הערך מוחזר ממנו.

3. ערך זה מוקצה למשתנה `square`.

אין צורך להקצות את התוצאה למשתנה, ניתן להציג אותה מיד:

```
def cylinder():
    r = float(input("Radius is"))
    h = float(input("Height is"))
    # square of the side of cylinder
    side = 2 * 3.14 * r * h
    # square of the base
    circle = 3.14 * r**2
    # total square of cylinder
    full = side + 2 * circle
    return full

print(cylinder())
```

```
Radius is1
Height is2
18.84
```

כאן המספר המתקבל מ- `cylinder()` מועבר ישירות לפונקציית ההדפסה. אם אנו פשוט כותבים `cylinder()` בתוכנית מבלי להקצות את הנתונים שהתקבלו למשתנה או להעביר אותם רחוק יותר, נתונים אלה יאבדו-אך לא תהיה שגיאת תחביר.

לפונקציה יכולות להיות הצהרות החזרה מרובות. עם זאת, רק אחד מהם מופעל תמיד, אליו מגיעה הפונקציה תוך כדי הביצוע. נניח שהחלטנו לעבד חריגה שמתרחשת במקרה של קלט שגוי. בענף למעט של מטפל בחריגות, הפונקציה יוצאת ללא חישובים ומעבירה את הערך:

```
def cylinder():
    try:
        r = float(input())
        h = float(input())
    except ValueError:
```

```
return
side = 2 * 3.14 * r * h
circle = 3.14 * r**2
full = side + 2 * circle
return full
print(cylinder())
```

במידה ונכניס תו במקום מספר נקבל החזרת חריגה. שימו לב שחריגה מחזירה אובייקט None. אובייקט זה שייך למחלקה (Class) NoneType. במידה שלפקודת return לא מוצמד ערך מסוים להחזרה, הפונקציה מחזירה את הערך None. (דומה לערך Null בשפת C).

```
/
None

1
2
18.84
```

יתר על כך. מוקדם יותר הסתכלנו על פונקציות שלא נראה שהחזירו ערך משום שלא הייתה להם פקודה "return". למען האמת היה ערך מוחזר, פשוט לא שמנו לב לזה, לא הקצנו שום משתנה ולא הצגנו אותו על המסך.

בפייתון כל פונקציה מחזירה משהו. אם אין לו פקודה "return", היא מחזירה ערך None. בדיוק כמו בפונקציה בעלת return "ריק".

## החזרת מספר ערכים מהפונקציה.

בשפות רבות (כמו שפת C, אסמבלי וכו') פונקציה יכולה להחזיר לא יותר מערך אחד. זה הבדל שנוצג לנו אפשרויות רבות – בעיקר לא להסתבך עם מצביעים כדי להחזיר מספר פרמטרים מהפונקציה. נראה את הדוגמה הקודמת בצורה שונה.

```
def cylinder():
    try:
        r = float(input())
        h = float(input())
    except ValueError:
        return
    side = 2 * 3.14 * r * h
    circle = 3.14 * r**2
    full = side + 2 * circle
    return circle,side,full

first,second, third=cylinder()
print("Side square is ", first)
print("Circle square is ", second)
print("Total square is ", third)
```

```
1
2
Side square is 3.14
Circle square is 12.56
Total square is 18.84
```

אפשרות הזאת של שפת פייתון נובעת מזה שאפשר להגדיר משתנים בצורה הבאה:

```
>>> a,b,c,d=1,2,3,4
>>> a
1
>>> b
2
>>> c
3
>>> d
4
```

הנקודה היא שכאשר אנו מגדירים משתנים בעזרת אופרטור ",", אנו בונים אובייקט שבשפת פייתון נקרא "tuple" - צירוף. כאשר מוגדר אובייקט מסוג זה, כל האיברים ששייכים לאובייקט מסודרים אחד אחרי שני. זה דומה מאוד למערך דינמי – מערך שגודלו לא מוגדר ולא מוגבל מראש. אם לאחר מילת return מופיע שם המשתנה היחיד או אובייקט כלשהו, הפונקציה מחזירה אותו באותה הצורה.

לא חובה להעביר פרמטרים למשתנים כדי להדפיס אותם – זה יכול לעבוד ישירות עם פקודת הדפסה:

```
def cylinder():
    try:
        r = float(input())
        h = float(input())
    except ValueError:
        return
    side = 2 * 3.14 * r * h
    circle = 3.14 * r**2
    full = side + 2 * circle
    return circle,side,full

print("Side square Circle square Total square ", cylinder())
```

```
1
2
Side square Circle square Total square (3.14, 12.56, 18.84)
```

פרמטרים וארגומנטים של פונקציות (arguments, parameters).

parameters

```
>>> def mathem (a, b):
        a = a / 2
        b = b + 10
        print (a + b)

>>> num1 = 100
>>> num2 = 12
>>> mathem (num1, num2)
1100.0
```

arguments

כאשר משתנים נמצאים במקום של הגדרה הם נקראים "פרמטרים" (parameters), כאשר משתנים נמצאים במקום של קריאה לפונקציה הם נקראים "ארגומנטים" (arguments).

בתכנות, פונקציות יכולות לא רק להחזיר נתונים, אלא גם לקבל אותם. זה עובד באמצעות הפרמטרים המצוינים בסוגריים בכותרת הפונקציה. מספר הפרמטרים יכול להיות כל אחד.

פרמטרים הם משתנים מקומיים שמוקצים להם ערכים בזמן שנקראת הפונקציה. הערכים הספציפיים המועברים לפונקציה כאשר היא נקראת ייקראו ארגומנטים. **יש לזכור כי יש מונחים שונים.** לדוגמה, "פרמטרים רשמיים" ו-"פרמטרים בפועל" בשפות אחרות. בפיתון הם נקראים בדרך כלל arguments.

```
>>> def mathem(a,b):
    a=a/2
    b=b+10
    print(a*b)

>>> num1=100
>>> num2=12
>>> mathem(num1,num2)
1100.0
```

כאשר נקראת פונקציה מועברים אליה ארגומנטים. הדוגמה מציגה את המשתנים הגלובליים num1 ו-num2. עם זאת, לא המשתנים הללו מועברים, אלא הערכים שלהם. במקרה זה המספרים הם 100 ו-12. במילים אחרות, נוכל לכתוב mathem(100,12) - לא יהיה הבדל.

כאשר המתורגמן ניגש לפונקציה כדי להתחיל את הביצוע, הוא מקצה את הארגומנטים שהועברו לפונקציה לפרמטרים המשתנים. לדוגמה למשתנה a יוקצה 100, ולמשתנה b יוקצה 12.

שינוי הערכים של a ו-b בגוף הפונקציה לא ישפיע על ערכי המשתנים num1 ו-num2 בשום דרך, הערכים ישארו זהים עבור השמות האלה. בפיתון התנהגות זו אופיינית לסוגי נתונים בלתי ניתנים להחלפה, הכוללים למשל מספרים ומיתרים. הם אומרים כי הנתונים מועברים לפי ערך לפונקציה. לכן, כאשר הוקצה ל-a המספר 100, זה כבר היה מספר שונה ומשתנה שונה. המספר 100 הועתק והוצב במיקום זיכרון נפרד עבור משתנה a.

ישנם סוגי נתונים הניתנים לשינוי. עבור פיתון למשל, רשימות ומילונים. במקרה זה, הנתונים מועברים באמצעות הפניה. קישור אליהם מועבר לפונקציה, ולא לנתונים עצמם. הקשר הזה קשור למשתנה מקומי. שינויים בנתונים כאלה באמצעות משתנה מקומי מתגלים כאשר ניגשים אליהם דרך המשתנה הגלובלי. זה עקב העובדה שמספר משתנים מתייחסים לאותם נתונים, לאזור זיכרון זהה.

הצורך לעבור בהתייחסות קשור בעיקר לחיסכון בזיכרון. בדרך כלל לא כדאי להעתיק סוגי נתונים מורכבים, שהם בעצם מבני נתונים. עם זאת, במידת הצורך, תמיד תוכלו לעשות זאת בכוח.

## כמות ארגומנטים משתנה.

שימו לב למשהו נוסף שלא היה בשפת C. בשפת C מספר ארגומנטים והפרמטרים צריך להיות זהה.

לא ניתן להעביר שלושה ארגומנטים אם הפונקציה מקבלת רק שניים. לא ניתן להעביר ארגומנט אחד אם הפונקציה דורשת שנים. בשפת פייתון זה עובד כמו בשפת C.

עם זאת, בפייתון, לפונקציות יש פרמטרים שכבר מוקצים להם ערך ברירת מחדל. במקרה זה, כשאתה מתקשר, לא ניתן להעביר ארגומנטים המתאימים לפרמטרים אלה. אם כי אפשר להעביר. ואז ערך ברירת המחדל יוחלף בערך המשודר.

```
def cylinder(h, r = 1):  
    side = 2 * 3.14 * r * h  
    circle = 3.14 * r**2  
    full = side + 2 * circle  
    return full  
figure1 = cylinder(4, 3)  
figure2 = cylinder(5)  
print(figure1)  
print(figure2)
```

```
131.88  
37.68
```

בקריאה השנייה לפונקציה cylinder אנו מציינים ארגומנט אחד בלבד, וזה יוקצה לפרמטר המשתנה h. המשתנה r יהיה 1.

על פי כללי התחביר של פייתון, בעת הגדרת הפונקציה, הפרמטרים שאליהם מוקצה ערך ברירת המחדל חייבים לעקוב אחר גב הפרמטרים שאין להם ערכי ברירת מחדל.

אבל כשאתה מפעיל פונקציה, ניתן לציין במפורש איזה ערך מתאים לאיזה פרמטר. במקרה זה, סדרם אינו משנה:

```
def cylinder(h, r = 1):  
    side = 2 * 3.14 * r * h  
    circle = 3.14 * r**2  
    full = side + 2 * circle  
    return full  
figure3 = cylinder(10, 2)  
figure4 = cylinder(r=2, h=10)  
print(figure3)  
print(figure4)
```

```
150.72  
150.72
```

במקרה זה, שתי קריאות הן קריאות עם ארגומנטים עם ערך זהים.

במקרה הראשון, מיפוי הפרמטרים למשתנים נמצא ברצף.  
במקרה השני, על ידי ארגומנטים, שהם שמות הפרמטרים.  
עוד דוגמה:

```
def oneOrMany(*a):  
    print(a)  
  
oneOrMany(1)  
oneOrMany('1',1, 2, 'abc')  
oneOrMany()
```

```
(1,)  
(1, 1, 2, 'abc')  
()
```

### שימוש במושג yield במקום מושג return.

הצהרת yield מעכבת את ביצוע הפונקציה ושולחת ערך בחזרה, אך לא מפסיקה את פעולת הפונקציה על מנת לאפשר לפונקציה להמשיך במקום בו היא נותרה ולשלוח מספר ערכים. לאחר חידוש, הפונקציה ממשיכה לבצע מיד לאחר ריצת התשואה האחרונה. זה מאפשר לקוד שלה לייצר סדרה של ערכים לאורך זמן, ולא לחשב אותם בבת אחת ולשלוח אותם כמו רשימה. Return שולח ערך מוגדר חזרה ואילו yield יכולה לייצר רצף ערכים. עלינו להשתמש ב-yield כשאנו רוצים לחזור על הרצף, אך לא רוצים לאחסן את כל הרצף בזיכרון.

yield משמשת בגנרטורים של פייתון. **פונקציית גנרטור מוגדרת כפונקציה רגילה, אך בכל פעם שהיא צריכה לייצר ערך, היא עושה זאת עם מילת המפתח yield-ל** ולא עם return. אם גוף המגן מכיל yield, הפונקציה הופכת אוטומטית לפונקציה של גנרטור. **בנושא הגנרטורים** נדון בהקדם.

```
A Simple Python program to demonstrate working of yield  
# A generator function that yields 1 for the first time,  
# 2 second time and 3 third time  
def simpleGeneratorFun():  
    yield 1  
    yield 2  
    yield 3  
  
# Driver code to check above generator function  
for value in simpleGeneratorFun():  
    print(value)
```

1  
2  
3

```
# A Python program to generate squares from 1
# to 100 using yield and therefore generator
# An infinite generator function that prints
# next square number. It starts with 1
def nextSquare():
    i = 1;
    # An Infinite loop to generate squares
    while True:
        yield i * i
        i += 1 # Next execution resumes
        # from this point
# Driver code to test above generator
# function
for num in nextSquare():
    if num > 100:
        break
    print(num,end=" ")
```

1 4 9 16 25 36 49 64 81 100

## גנרטורים של פייתון.

ישנם 2 סוגי גנרטורים:

פונקציה - גנרטור.

פונקציית גנרטור מוגדרת כפונקציה רגילה, אך בכל פעם שהיא צריכה לייצר ערך, היא עושה זאת באמצעות מילת המפתח yield ולא עם return. אם גוף המגן מכיל yield, הפונקציה הופכת אוטומטית לפונקציה של גנרטור.

גנרטור – אובייקט.

פונקציות גנרטור מחזירות אובייקט גנרטור. אובייקטים של גנרטורים משמשים באמצעות קריאה לשיטה הבאה באובייקט הגנרטור או באמצעות אובייקט הגנרטור בלולאה "for in" (כפי שמוצג בתוכנית לעיל).

אובייקט	פונקציה
<pre># A Python program to demonstrate use of # generator object with next() # A generator function def simpleGeneratorFun():     yield 1</pre>	<pre># A generator function that yields # 1 for first time, # 2 second time and 3 third time def simpleGeneratorFun():     yield 1</pre>



<pre> yield 2 yield 3 # x is a generator object x = simpleGeneratorFun()  # Iterating over the generator # object using next print(x.__next__()) print(x.__next__()) print(x.__next__()) </pre>	<pre> yield 2 yield 3  # Driver code to check above # generator function for value in simpleGeneratorFun():     print(value) </pre>
<pre> 1 2 3 </pre>	<pre> 1 2 3 </pre>

תרגול בנושא פונקציות:

תרגול בנושא טבלאות מעקב:

תרגיל מס' 1.

תפענח את הקוד הבא: מה עושה הפונקציה, מה מבצעת התוכנית. תן מספר דוגמאות לפלט שונה בהתאם לקלט שונה.

```

def average(n1, n2):
    m = (n1 + n2) / 2
    return m
a = int(input("A = "))
b = int(input("B = "))
avrg = average(a, b)
print(round(avrg, 2))

```

תרגיל מס' 2.

תפענח את הקוד הבא: מה מבצעת הפונקציה?, מה עושה תוכנית. תן מספר דוגמאות לפלט שונה בהתאם לקלט שונה.

```

import random
def TriangleInequality(A,B,C):
    return (A < B+C) and (B < A+C) and (C < A+B)
a,b,c = [random.randrange(1, 6) for i in range(0,3)]
while not TriangleInequality(a,b,c):
    a,b,c = [random.randrange(1, 6) for i in range(0,3)]

print("Side a: ", a)
print("Side b: ", b)

```

```
print("Side c: ", c)
bool_expr = (a == b and a == c)
print("Equal sides: ",bool_expr)
```

תרגיל מס' 3.

תפענח את הקוד הבא: מה מבצעת הפונקציה ?, מה עושה תוכנית. תן מספר דוגמאות לפלט שונה בהתאם לקלט שונה.

```
import random
def TriangleInequality(A,B,C):
    return (A < B+C) and (B < A+C) and (C < A+B)
a,b,c = [random.randrange(1, 6) for i in range(0,3)]
while not TriangleInequality(a,b,c):
    a,b,c = [random.randrange(1, 6) for i in range(0,3)]

print("Side a: ", a)
print("Side b: ", b)
print("Side c: ", c)

bool_expr = (a == b or a == c or b == c)
print("Equal borders: ",bool_expr)
```

תרגיל מס' 4.

תפענח את הקוד הבא: מה מבצעת הפונקציה ?, מה עושה תוכנית. תן מספר דוגמאות לפלט שונה בהתאם לקלט שונה.

```
import random
def TriangleInequality(A,B,C):
    return (A < B+C) and (B < A+C) and (C < A+B)

a,b = [random.randrange(1, 7) for i in range(0,2)]
c = random.randrange(5, 12)
while not TriangleInequality(a,b,c):
    a,b = [random.randrange(1, 7) for i in range(0,2)]
    c = random.randrange(5, 12)
#a,b,c = [5,3,4]
print("Triangle")
print("Side a: ", a)
print("Side b: ", b)
```

```
print("Side c: ", c)
bool_expr = ((a*a == b*b + c*c) or (b*b == a*a + c*c) or (c*c == b*b + a*a))
print("90 degrees : ",bool_expr)
```

תרגיל מס' 5.

תפענח את הקוד הבא: מה מבצעת הפונקציה ?, מה עושה תוכנית. תן מספר דוגמאות לפלט שונה בהתאם לקלט שונה.

```
import random
def TriangleInequality(A,B,C):
    return (A < B+C) and (B < A+C) and (C < A+B)
a,b,c = [random.randrange(1, 11) for i in range(0,3)]
print("Triangle")
print("Side a: ", a)
print("Side b: ", b)
print("Side c: ", c)
bool_expr = (TriangleInequality(a,b,c))
print("Triangle a, b, c: ",bool_expr)
```

תרגיל מס' 6.

תפענח את הקוד הבא: מה מבצעת הפונקציה ?, מה עושה תוכנית. תן מספר דוגמאות לפלט שונה בהתאם לקלט שונה.

```
def print_pascal_triangle(size):
    for i in range(0, size):
        for j in range(0, i + 1):
            print(decide_number(i, j), end=" ")
        print()

def decide_number(n, k):
    num = 1
    if k > n - k:
        k = n - k
    for i in range(0, k):
        num = num * (n - i)
        num = num // (i + 1)
    return num

rows = 7
print_pascal_triangle(rows)
```

תפענח את הקוד הבא: מה מבצעת הפונקציה ?, מה עושה תוכנית. תן מספר דוגמאות לפלט שונה בהתאם לקלט שונה.

```
def ones(B):
    match B:
        case 1:
            print("One",end=' ')
        case 2:
            print("Two",end=' ')
        case 3:
            print("Three",end=' ')
        case 4:
            print("Four",end=' ')
        case 5:
            print("Five",end=' ')
        case 6:
            print("Six",end=' ')
        case 7:
            print("Seven",end=' ')
        case 8:
            print("Eight",end=' ')
        case 9:
            print("Nine",end=' ')
def teens(B):
    match B:
        case 2:
            print("Twen",end="")
        case 3:
            print("Thir",end="")
        case 4:
            print("Four",end="")
        case 5:
            print("Fiv",end="")
        case 6:
            print("Six",end="")
        case 7:
            print("Seven",end="")
        case 8:
            print("Eigh",end="")
        case 9:
            print("Nin",end="")
A=int(input("Enter the number: "))
```

```

if 10000>A>999:
    B=A//1000
    ones(B)
    print("Thousand ",end=' ')
    A%=1000
if 1000>A>99:
    B = A // 100
    ones(B)
    print("Hundred ", end=' ')
    A %= 100
if A==12:
    print("Twelve ", end=' ')
elif A==11:
    print("Eleven ", end=' ')
elif A==10:
    print("Twelve ", end=' ')
else:
    if 100>A>=20:
        B = A // 10
        teens(B)
        print("ty",end=' ')
        A %= 10
        ones(A)
    elif 20>A>12:
        B = A // 10
        teens(B)
        print("teen", end=' ')
    else:
        A%=10
        ones(A)

```

תרגיל מס' 8.

תפענח את הקוד הבא: מה מבצעת הפונקציה ?, מה עושה תוכנית. תן מספר דוגמאות לפלט שונה בהתאם לקלט שונה.

```

def fibonacci(n):
    if n in (1, 2):
        return 1
    return fibonacci(n - 1) + fibonacci(n - 2)
n = int(input())
print(fibonacci(n))

```

תרגיל מס' 9.

תפענח את הקוד הבא: מה מבצעת הפונקציה ?, מה עושה תוכנית. תן מספר דוגמאות לפלט שונה בהתאם לקלט שונה.

```
def average(*n):
    m = sum(n) / len(n)
    return m
avrg1 = average(10, 12)
avrg2 = average(1, 2, 3.4, 3, 8)
avrg3 = average(-0.5, 2.3, 1, 5.3, -2, 1.55)
print(round(avrg1, 2))
print(round(avrg2, 2))
print(round(avrg3, 2))
```

תרגיל מס' 10.

תפענח את הקוד הבא: מה מבצעת הפונקציה ?, מה עושה תוכנית. תן מספר דוגמאות לפלט שונה בהתאם לקלט שונה.

```
def list_avrg(lst):
    l = len(lst)
    suma = 0
    for i in lst:
        suma += i
    return suma / l

print("Input integers:")
a = input()
a = a.split()
for i in range(len(a)):
    a[i] = int(a[i])
avrg = list_avrg(a)
print("Average:")
print(round(avrg, 2))
```

תרגיל מס' 11.

תפענח את הקוד הבא: מה מבצעת הפונקציה ?, מה עושה תוכנית. תן מספר דוגמאות לפלט שונה בהתאם לקלט שונה.

```
def fib_row(item):
    f1 = 0
    f2 = 1
    print(f1, f2, end=' ')
```

```

while item > 2:
    buff = f2
    f2 = f1 + f2
    f1 = buff
    print(f2, end=' ')
    item -= 1
print()

n = int(input())
fib_row(n)

```

תרגיל מס' 12.

תפענח את הקוד הבא: מה מבצעת הפונקציה ?, מה עושה תוכנית. תן מספר דוגמאות לפלט שונה בהתאם לקלט שונה.

```

def factorial(a):
    if a == 1:
        return a
    return a * factorial(a-1)

n = int(input())
print(factorial(n))

```

תרגיל מס' 13.

תפענח את הקוד הבא: מה מבצעת הפונקציה ?, מה עושה תוכנית. תן מספר דוגמאות לפלט שונה בהתאם לקלט שונה.

```

from math import sqrt
def is_prime(n):
    if n < 2:
        return False
    if n == 2:
        return True
    limit = sqrt(n)
    i = 2
    while i <= limit:
        if n % i == 0:
            return False
        i += 1
    return True

for i in range(3):
    num = int(input())
    print(is_prime(num))

```

תרגיל מס' 14.

תפענח את הקוד הבא: מה מבצעת הפונקציה ?, מה עושה תוכנית. תן מספר דוגמאות לפלט שונה בהתאם לקלט שונה.

```
def lcm(a, b):
    m = a * b
    while a != 0 and b != 0:
        if a > b:
            a %= b
        else:
            b %= a
    return m // (a + b)

while 1:
    try:
        x = int(input('a = '))
        y = int(input('b = '))
        print("LCM:", lcm(x, y))
    except ValueError:
        break
```

תרגיל מס' 15.

תפענח את הקוד הבא: מה מבצעת הפונקציה ?, מה עושה תוכנית. תן מספר דוגמאות לפלט שונה בהתאם לקלט שונה.

```
def lcm(a, b):
    import math
    return (a * b) // math.gcd(a, b)

while 1:
    try:
        x = int(input('a = '))
        y = int(input('b = '))
        print("LCM:", lcm(x, y))
    except ValueError:
        break
```

תרגיל מס' 16.

תפענח את הקוד הבא: מה מבצעת הפונקציה ?. תן מספר דוגמאות לפלט שונה בהתאם לקלט שונה.

```
def to_float(num):
    if isinstance(num, (int, float)):
```



```
return float(num)
return "Impossible"
```

תרגיל מס' 17.

תפענח את הקוד הבא: מה מבצעת הפונקציה? תן מספר דוגמאות לפלט שונה בהתאם לקלט שונה.

```
def mul_to_int(a, b):
    res = a * b
    if float(res).is_integer():
        return int(res)
    return res
```

תרגיל מס' 18.

תפענח את הקוד הבא: מה מבצעת הפונקציה? תן מספר דוגמאות לפלט שונה בהתאם לקלט שונה.

```
def round_standard(num):
    if num >= 0:
        sign = 1
    else:
        sign = -1
    return sign * int((abs(num) + 0.5))
```

תרגיל מס' 19.

תפענח את הקוד הבא: מה מבצעת הפונקציה? תן מספר דוגמאות לפלט שונה בהתאם לקלט שונה.

```
def eqv(a, b, c):
    res = a + b
    e = 0.01 / 100
    tolerance = e * max(abs(a), abs(b))
    return abs(res - c) <= tolerance
```

תרגיל מס' 20.

תפענח את הקוד הבא: מה מבצעת הפונקציה? תן מספר דוגמאות לפלט שונה בהתאם לקלט שונה.

```
def eqv(a, b, c):
    from math import isclose
    return isclose(a + b, c, rel_tol=0.01 / 100, abs_tol=0)
```

תרגיל מס' 21.

תפענח את הקוד הבא: מה מבצעת הפונקציה? תן מספר דוגמאות לפלט שונה בהתאם לקלט שונה.

```
def factorial(n):  
    res = 1  
    for i in range(1, n + 1):  
        res *= i  
    return res
```

תרגיל מס' 22.

תפענח את הקוד הבא: מה מבצעת הפונקציה? תן מספר דוגמאות לפלט שונה בהתאם לקלט שונה.

```
def func1(*args):  
    for i in args:  
        print(i)
```

תרגיל מס' 23.

תפענח את הקוד הבא: מה מבצעת הפונקציה? תן מספר דוגמאות לפלט שונה בהתאם לקלט שונה.

```
def calculation(a, b):  
    addition = a + b  
    subtraction = a - b  
    return addition, subtraction
```

תרגיל מס' 24.

תפענח את הקוד הבא: מה מבצעת הפונקציה? מה עושה תוכנית. תן מספר דוגמאות לפלט שונה בהתאם לקלט שונה.

```
def show_employee(name, salary=9000):  
    print("Name:", name, "salary:", salary)
```

תרגיל מס' 25.

תפענח את הקוד הבא: מה מבצעת הפונקציה? תן מספר דוגמאות לפלט שונה בהתאם לקלט שונה.

```
def outer_fun(a, b):  
    square = a ** 2
```

```
def addition(a, b):
    return a + b
add = addition(a, b)
return add + 5
```

תרגיל מס' 26.

תפענח את הקוד הבא: מה מבצעת הפונקציה ?, מה עושה תוכנית. תן מספר דוגמאות לפלט שונה בהתאם לקלט שונה.

```
def AddLeftDigit(D):
    global K
    n = len(str(K))
    K = D * 10 ** n + K
K = random.randrange(100, 1000)
print("Number K: ", K)
for i in range(3):
    D = random.randrange(0, 10)
    print("Число D, ", i + 1, ": ", D)
    AddLeftDigit(D)
print("Changed K: ", K)
print()
```

תרגיל מס' 27.

תפענח את הקוד הבא: מה מבצעת הפונקציה ?, מה עושה תוכנית. תן מספר דוגמאות לפלט שונה בהתאם לקלט שונה.

```
import random

def Sign(X):
    if X < 0:
        return -1
    elif X > 0:
        return 1
    return 0

A = random.randrange(-10,10)
B = random.randrange(-10,10)
print("A = ", A)
print("B = ", B)
s_A = Sign(A)
s_B = Sign(B)
print("Sign(A) = ", s_A)
print("Sign(B) = ", s_B)
print("Sign(A) + Sign(B) = ", s_A+s_B)
```

תרגיל מס' 28.

תפענח את הקוד הבא: מה מבצעת הפונקציה? תן מספר דוגמאות לפלט שונה בהתאם לקלט שונה: def Calc(A,B,Op):

```
if Op == 1:
    return A - B
if Op == 2:
    return A * B
if Op == 3:
    return A / B
return A + B
```

תרגיל מס' 29.

תפענח את הקוד הבא: מה מבצעת הפונקציה? מה עושה תוכנית. תן מספר דוגמאות לפלט שונה בהתאם לקלט שונה.

```
import random
def Even(K):
    if K%2 == 0:
        return True
    return False

s = 0
for i in range(0,10):
    x = random.randrange(1,11)
    print(x,end="; ")
    s += int(Even(x))
print("\nAmount of even numbers:",s)
```

תרגיל מס' 30.

תפענח את הקוד הבא: מה מבצעת הפונקציה? מה עושה תוכנית. תן מספר דוגמאות לפלט שונה בהתאם לקלט שונה.

```
import random
import math
def IsSquare(K):
    x = int(math.sqrt(K))
    if K == x*x:
        return True
    return False

s = 0
```

```

for i in range(0,10):
    x = random.randrange(1,101)
    print(x,end="; ")
    s += int(IsSquare(x))

print("\nAmount of Squares:",s)

```

תרגיל מס' 31.

תפענח את הקוד הבא: מה מבצעות הפונקציות. תן מספר דוגמאות לפלט שונה בהתאם לקלט שונה.

```

def IsPowerN(K,N):
    x = int(round(math.log(K,N)))
    if K == N**x:
        return True
    return False

def IsPowerNa(K,N):
    while K > 1:
        K /= N
    if K == 1.0:
        return True
    return False

```

תרגיל מס' 32.

תפענח את הקוד הבא: מה מבצעת הפונקציה ?, מה עושה תוכנית. תן מספר דוגמאות לפלט שונה בהתאם לקלט שונה.

```

import random
import math

def IsPrime(N):
    if N % 2 == 0 and N > 2:
        return False
    for i in range(3, int(math.sqrt(N)) + 1, 2):
        if N % i == 0:
            return False
    return True

s = 0
for i in range(0,10):
    x = random.randrange(1,1000)
    s += int(IsPrime(x))

```

```
print(x,":",IsPrime(x))
print("\nAmount of IsPrime:",s)
```

תרגיל מס' 33.

תפענח את הקוד הבא: מה מבצעת הפונקציה ?, מה עושה תוכנית. תן מספר דוגמאות לפלט שונה בהתאם לקלט שונה.

```
import random
import math

def DigitCount(N):
    i = 0
    q = N
    while q >= 1:
        q /= 10
        i += 1
    return i

for i in range(0,10):
    x = random.randrange(1,100)
    print(x,":",DigitCount(x))
```

תרגיל מס' 34.

תפענח את הקוד הבא: מה מבצעת הפונקציה ?, מה עושה תוכנית. תן מספר דוגמאות לפלט שונה בהתאם לקלט שונה.

```
import random
import math

def DigitN(K,N):
    q = K
    i = 0
    while q >= 1:
        r = q % 10
        q = int(q/10)
        i += 1
        if i == N:
            return r
    return -1

for i in range(0,10):
    x = random.randrange(1,10000)
```

```
N = random.randrange(1,6)
print(x,":",N,":",DigitN(x,N))
```

תרגיל מס' 35.

תפענח את הקוד הבא: מה עושה כל פונקציה שיש בתוכנית. תן מספר דוגמאות לפלט שונה בהתאם לקלט שונה.

```
import random
def Palindrom(K):
    return int(str(K) + str(K)[::-1])
def IsPalindrom(K):
    result = False
    num = str(K)
    try:
        val = int(num)
        if num == str(num)[::-1]:
            result = True
    except ValueError:
        print("That's not a valid number, Try Again !")
    return result
def IsPalindrom2(n):
    temp=n
    rev=0
    while(n>0):
        dig=n%10
        rev=rev*10+dig
        n=n//10
    if(temp==rev):
        result = True
    else:
        result = False
    return result
```

תרגיל מס' 36.

תפענח את הקוד הבא: מה מבצעת הפונקציה? תן מספר דוגמאות לפלט שונה בהתאם לקלט שונה.

```
def Fib(N):
    f1 = 1
    f2 = 1
    if N <= 2:
        return 1
    while N > 2:
```

```

F = f1 + f2
f1 = f2
f2 = F
N -= 1
return F

```

תרגיל מס' 37.

תפענח את הקוד הבא: מה מבצעת הפונקציה? תן מספר דוגמאות לפלט שונה בהתאם לקלט שונה.

```

def Power2(A,N):
    i = 0
    p = 1

    if N < 0:
        N1 = -N
    else:
        N1 = N

    while i < N1:
        p *= A
        i += 1

    if N < 0:
        p = 1 / p

    return p

```

תרגיל מס' 38.

תפענח את הקוד הבא: מה מבצעת הפונקציה?, מה עושה תוכנית. תן מספר דוגמאות לפלט שונה בהתאם לקלט שונה.

```

def Exp1(x,eps):
    if eps <= 0:
        print("Epsilon should be greater than 0")
    y = x
    exp = 1.0 + x
    i = 2
    while abs(y) > eps:
        y *= x / i
        i += 1
        exp += y
    return exp

```



תרגיל מס' 39.

תפענח את הקוד הבא: מה מבצעת הפונקציה ?, מה עושה תוכנית. תן מספר דוגמאות לפלט שונה בהתאם לקלט שונה.

```
import random
import math

def Sin1(x,eps):
    if eps <= 0:
        print("Epsilon should be greater than 0")
    y = x
    f = x
    i = 3
    while abs(y) > eps:
        y *= (-1) * x * x / ((i-1)*i)
        i += 2
        f += y
    return f
eps = 0.01
for i in range(0, 6):
    x = math.pi / 4
    print("eps = ", eps, "; sin(",x,") = ",Sin1(x,eps),";",math.sin(x))
eps /= 10
```

תרגיל מס' 40.

תפענח את הקוד הבא: מה מבצעת הפונקציה ?, מה עושה תוכנית. תן מספר דוגמאות לפלט שונה בהתאם לקלט שונה.

```
import random
import math

def Cos1(x,eps):
    if eps <= 0:
        print("Epsilon should be greater than 0")
    y = 1
    f = 1
    i = 2
    while abs(y) > eps:
        y *= (-1) * x * x / ((i-1)*i)
        i += 2
        f += y
    return f
```

```

eps = 0.01
for i in range(0, 6):
    x = math.pi / 4
    print("cos(", x, ") = ", Cos1(x, eps), ";", math.cos(x))
    eps /= 10

```

תרגיל מס' 41.

תפענח את הקוד הבא: מה מבצעת הפונקציה ?, מה עושה תוכנית. תן מספר דוגמאות לפלט שונה בהתאם לקלט שונה.

```

import random
import math

def NOD2(A,B):
    while B != 0:
        A,B = B,A%B
    return A

for i in range(0,4):
    A = random.randrange(1,1000)
    B = random.randrange(1,1000)
    print(A, ";", B, ";", NOD2(A,B))

```

תרגיל מס' 42.

תפענח את הקוד הבא: מה מבצעת הפונקציה ?, מה עושה תוכנית. תן מספר דוגמאות לפלט שונה בהתאם לקלט שונה.

```

import random
import math

def NOD2(A,B):
    while B != 0:
        A,B = B,A%B
    return A

def NOK2(A,B):
    return A * B / NOD2(A,B)

for i in range(0,4):

```

```

A = random.randrange(1,100)
B = random.randrange(1,100)

y = NOK2(A,B)
print(A,";",B,";",y)
print(y,":",A,"=",y/A)
print(y,":",B,"=",y/B)
print()

```

תרגיל מס' 43.

תפענח את הקוד הבא: מה מבצעת הפונקציה ?, מה עושה תוכנית. תן מספר דוגמאות לפלט שונה בהתאם לקלט שונה.

```

import random
import math

def NOD2(A,B):
    while B != 0:
        A,B = B,A%B
    return A

def NOD3(A,B,C):
    return NOD2(NOD2(A,B),C)

for i in range(0,10):
    A = random.randrange(1,100)
    B = random.randrange(100,300)
    C = random.randrange(300,500)

    y = NOD3(A,B,C)
    print(A,";",B,";",C,";",y)
    print()

```

תרגול בנושא כתיבת קוד.

תרגיל מס' 1.

כתוב פונקציה שמקבלת 2 מספרים מהמשתמש ומחזירה את הממוצע שלהם.

תרגיל מס' 2.

כתוב פונקציה שמקבלת מספר כלשהו של מספרים מהמשתמש ומחזירה את הממוצע שלהם.

תרגיל מס' 3.

כתוב פונקציה  $distance(x1, y1, x2, y2)$  שמקבלת מהמשתמש קואורדינטות של 2 נקודות ומחזירה את המרחק ואת אמצע קטע.

תרגיל מס' 4.

כתוב פונקציה  $min4(a, b, c, d)$  שמקבלת 4 מספרים ומחזירה את המספר הקטן ביותר.

תרגיל מס' 5.

כתוב פונקציה שמקבלת מספר של איבר של סדרת פיבונצ'י. הפונקציה מדפיסה את סדרת הפיבונצ'י לפי כמות איברים שפונקציה מקבלת.

תרגיל מס' 6.

כתוב פונקציה שמחשבת עצרת בעזרת רקורסיה.

תרגיל מס' 7.

כתוב פונקציה שבודקת האם מספר כלשהו הוא מספר ראשוני או לא ותדפיס תגובה מתאימה.

תרגיל מס' 8.

כתוב פונקציה שמחשבת את המספר הקטן ביותר שמתחלק ב- 2 מספרים שנקלטים מהמשתמש.

חיפוש של המספר הקטן ביותר שמתחלק בשני מספרים הנתונים: (כפולה משותפת מינימלית) מצא את הכפל הנפוץ הקטן ביותר (LCM) של זוג מספרים לפי הנוסחה

$$LCM(a,b) = ab / GCD$$

(least common multiple) (Greatest common divisor)

כאשר  $a$  ו- $b$  הם מספרים טבעיים,  $GCD$  הוא הגורם השכיח הגדול ביותר.

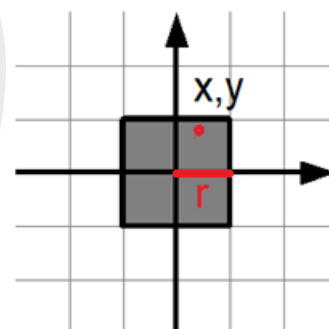
מתנאי הבעיה ברור שכדי למצוא את ה-LCM, יש למצוא תחילה את ה-GCD. ניתן לחשב את האחרון על ידי מציאת השארית חלוקת המספר הגדול יותר מהזוג בהדרגה, על ידי אחד קטן יותר והקצאת השארית למשתנה הקשור למספר גדול (ראו האלגוריתם האוקלידס). בשלב מסוים הערך של אחד המשתנים יהפוך ל 0. כשזה יקרה, השני יכיל GCD. אם לא ידוע איזה משתנה מכיל את ה-GCD, ניתן פשוט להוסיף את הערכים של שני המשתנים.

בקוד שלמטה משתמשים בפונקציה כדי למצוא את ה-LCM, שלוקח שני מספרים ומחזיר את הכפולה הנפוצה ביותר שנמצאה.

בענף הראשי של התוכנית, הפונקציה נקראת בלולאה, שמסתיימת אם לא ניתן להמיר את מה שהוזן למספר שלם. במקרה זה, נזרקת חריגה ותוכנית הולכת לענף חוץ.

תרגיל מס' 9.

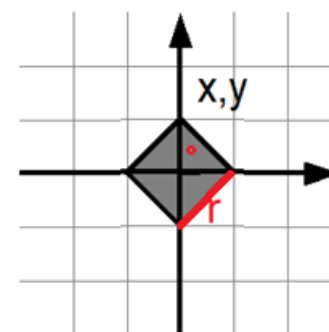
כתוב פונקציה  $IsPointInSquare(x, y, r)$  שמקבלת פרמטרים לפי ציור הבא:



פונקציה בודקת האם נקודה שייכת לריבוע.

תרגיל מס' 10.

כתוב פונקציה  $IsPointInSquare(x, y, r)$  שמקבלת פרמטרים לפי ציור הבא:



פונקציה בודקת האם נקודה שייכת לריבוע.

תרגיל מס' 11.

כתוב פונקציית `getInput` ללא פרמטרים, מבקשת קלט מהמקלדת ומחזירה את המחרוזת שהתקבלה לתוכנית הראשית.

תרגיל מס' 12.

כתוב פונקציית `testInput` עם פרמטר אחד. בגופה נמצאת בדיקה האם ניתן להמיר את הערך המועבר אליה למספר שלם. אם אפשר, מחזיר `True`, אם לא `False`.

תרגיל מס' 13.

כתוב פונקציית `strToInt()` מכילה פרמטר אחד. הפונקציה ממירה את הערך שהועבר לסוג שלם, ומחזירה את המספר שהתקבל.

תרגיל מס' 14.

כתוב פונקציית `printInt()` מכילה פרמטר אחד. היא מציגה את הערך שהועבר על המסך ולא מחזירה כלום.

תרגיל מס' 15.

כתוב פונקציה `IsPointInCircle(x, y, xc, yc, r)` שמקבלת מרכז המעגל, רדיוס המעגל וקואורדינטות של הנקודה. פונקציה בודקת האם נקודה שייכת למעגל ומחזירה תוצאה: `TRUE` או `FALSE`.

תרגיל מס' 16.

כתוב פונקציה בשם `ReduceFraction(n, m)`, שמטרת לצמצם את השבר  $\frac{n}{m}$ , ככל שאפשר ולהדפיס את תוצאת הצמצום. לדוגמה: קלט - 6,9 פלט 2,3.

תרגיל מס' 17.

כתוב פונקציה `MinDivisor(n)` שמקבלת מספר שלם ומחזירה מחלק הקטן ביותר שלו (גדול מ-1).

תרגיל מס' 18.

כתוב פונקציה `MaxDivisor(n)` שמקבלת מספר שלם ומחזירה מחלק הגדול ביותר שלו.

תרגיל מס' 19.

כתוב פונקציה `IsPrime(x)` שבודקת האם מספר הוא מספר ראשוני או לא. פונקציה מחזירה `TRUE` או `FALSE`.

תרגיל מס' 20.

כתוב פונקציה  $\text{power}(a, n)$  שמקבלת בסיס וחזקה ומחזירה את התוצאה (פונקציה מבצעת פעולה בעזרת לולאה).

תרגיל מס' 21.

כתוב פונקציה שמקבלת נתונים שלמים עד לקבלת 0. פונקציה מחזירה סכום של הסדרה.

תרגיל מס' 22.

כתוב פונקציה שמקבלת נתונים שלמים עד לקבלת 0. פונקציה מדפיסה את המחרוזת בסדר הפוך.

תרגיל מס' 23.

כתוב פונקציות שמבצעת חישוב של סינוס הזווית לפי נוסחה:

$$\sin(x) = x - x^3/(3!) + x^5/(5!) - \dots + (-1)^n \cdot x^{2n+1}/((2 \cdot n + 1)!) + \dots$$

פונקציה שמקבלת זווית במעלות, מעבירה אותו לרדיאנים.

פונקציה לחישוב  $n!$ .

פונקציה שמקבלת זווית ברדיאנים, כמות איברים לחישוב ומחזירה תוצאת הסינוס.

תרגיל מס' 24.

כתוב פונקציות שמבצעת חישוב של קוסינוס הזווית לפי נוסחה:

$$\cos(x) = 1 - x^2/(2!) + x^4/(4!) - \dots + (-1)^n \cdot x^{2n}/((2 \cdot n)!) + \dots$$

פונקציה שמקבלת זווית במעלות, מעבירה אותו לרדיאנים.

פונקציה לחישוב  $n!$ .

פונקציה שמקבלת זווית ברדיאנים, כמות איברים לחישוב ומחזירה תוצאת הקוסינוס.

תרגיל מס' 25.

כתוב פונקציה לחישוב של מחלק משותף של שני מספרים.  
בעזרת פונקציה הזאת כתוב פונקציה לחישוב מחנה משותף של 3 מספרים.

תרגיל מס' 26.

כתוב פונקציה שמקבלת כמות שניות שעברו מאירוע מסוים ומחזירה את כמות שעות, דקות ושניות ושעברו.

תרגיל מס' 26.

כתוב פונקציה שמקבלת תו ASCII אקראי ומחזירה תוצאה FALSE OR TRUE, האם תו הוא אות.

תרגיל מס' 27.

כתוב פונקציה שמקבלת תו ASCII אקראי ומחזירה תוצאה FALSE OR TRUE, האם תו הוא אות. אם כן האם היא אות גדולה או קטנה.



## פרק ו'. רשימות. Lists.

### הגדרת רשימות.

רשימה בפייתון היא מחלקת נתונים מובנת, שהיא אחד מסוגי מבני הנתונים. ניתן לייצג את מבנה הנתונים כיחידה מורכבת, המאחדת קבוצה של מבנה נתונים פשוטים יותר. לכל סוג של מבנה נתונים מאפיינים משלו. רשימה היא רצף הניתן לשינוי של אלמנטים שרירותיים. לרוב המוחלט של שפות תכנות אחרות יש סוג נתונים כה נרחב כמו מערך. לפייתון אין סוג מובנה כזה. עם זאת, רשימות הן מערכים למעשה עם יוצא מן הכלל אחד. האלמנטים המרכיבים את המערך חייבים להיות שייכים לאותו סוג נתונים, עבור רשימות פייתון אין הגבלה כזו.

לדוגמה, מערך עשוי להכיל מספרים שלמים בלבד או רק מספרים ממעשיים או רק תווים. רשימה יכולה להכיל גם אלמנטים מאותו סוג, מה שהופך אותה זהה למערך. אבל מקובל בהחלט שרשימה אחת מכילה מספרים ותווים כאחד, כמו גם משהו אחר. ישנן דרכים רבות ליצור רשימות. בואו ניצור אותו על ידי ספירה פשוטה של אלמנטים:

```
>>> a = [12, 3.85, "black", -4]
>>> a
[12, 3.85, 'black', -4]
```

אז יש לנו רשימה שהוקצתה למשתנה a. בפייתון רשימה מוגדרת על ידי סוגריים מרובעים. היא מכילה ארבעה אלמנטים. אם איפשהו בתוכנית אנו זקוקים לרשימה כולה, נקבל גישה אליה על ידי ציון משתנה אחד בלבד, ובדוגמה הזו מדובר במשתנה a.

הפריטים ברשימה מסודרים, זה משנה באיזה סדר הם מסודרים. לכל רכיב אינדקס או מספר משלו. האינדקס מתחיל מהתחלה. במקרה זה, למספר 12 יש אינדקס של 0, המחוזת "black" היא אינדקס של 2. כדי לחלץ אלמנט ספציפי, יש לציין את אינדקס האלמנט הדרוש בסוגריים מרובעים אחרי שם המשתנה:

```
>>> a[0]
12
>>> a[3]
-4
```

בשפת פייתון יש אפשרות מעניינת – לתת אינדקס מהסוף להתחלה ע"י אינדקסים שליליים.

```
>>> a[-1]
-4
>>> a[-2]
'black'
>>> a[-3],a[-4]
(3.85, 12)
```

```
>>> a[0:2]
[12, 3.85]
>>> a[:3]
[12, 3.85, 'black']
>>> a[2:]
['black', -4]
>>> a[:]
[12, 3.85, 'black', -4]
```

## פעולות בסיסיות עם רשימות.

ניתן לחבר רשימות ולעתיק רשימות אחד לתוך שנייה. במידה ומכפילים רשימה בערך שלם קבוע - זה פשוט משכפל את הרשימה מספר פעמים.

```
>>> a=['dis',1,3.4,"qwerty",-3]
>>> b=[2,3,1,"qw"]
>>> c=a+b
>>> a
['dis', 1, 3.4, 'qwerty', -3]
>>> b
[2, 3, 1, 'qw']
>>> c
['dis', 1, 3.4, 'qwerty', -3, 2, 3, 1, 'qw']
>>> d=a
>>> d
['dis', 1, 3.4, 'qwerty', -3]
>>> a=c*2
>>> a
['dis', 1, 3.4, 'qwerty', -3, 2, 3, 1, 'qw', 'dis', 1, 3.4, 'qwerty', -3, 2, 3, 1, 'qw']
>>> b = [1, 2, 3, 4, 5, 6]
>>> a=b[:2]+b[3:]
>>> a
[1, 2, 4, 5, 6]
>>> a[0:2]='q','w','z'
>>> a
['q', 'w', 'z', 4, 5, 6]
>>> b
[1, 2, 3, 4, 5, 6]
>>> b[0:2]="qwerty"
>>> b
['qwerty', 3, 4, 5, 6]
```

```
>>> a=a/2
Traceback (most recent call last):
  File "<pyshell#80>", line 1, in <module>
    a=a/2
TypeError: unsupported operand type(s) for /: 'list' and 'int'
>>> a=a*1.2
Traceback (most recent call last):
  File "<pyshell#81>", line 1, in <module>
    a=a*1.2
TypeError: can't multiply sequence by non-int of type 'float'
>>> a=a-a
Traceback (most recent call last):
  File "<pyshell#82>", line 1, in <module>
    a=a-a
TypeError: unsupported operand type(s) for -: 'list' and 'list'
>>> a=a-2
Traceback (most recent call last):
  File "<pyshell#83>", line 1, in <module>
    a=a-2
TypeError: unsupported operand type(s) for -: 'list' and 'int'
>>> a=a+2
Traceback (most recent call last):
  File "<pyshell#84>", line 1, in <module>
    a=a+2
TypeError: can only concatenate list (not "int") to list
>>> a=a+"2"
Traceback (most recent call last):
  File "<pyshell#85>", line 1, in <module>
    a=a+"2"
TypeError: can only concatenate list (not "str") to list
```

## פונקציות נוספות בנושא רשימות.

כדי לראות רשימת פונקציות נוספות יש להשתמש בפקודת `dir()`

```
>>> dir(list)
['_add_', '_class_', '_contains_', '_delattr_', '_delitem_', '_dir_', '_doc_',
'_eq_', '_format_', '_ge_', '_getattr_', '_getitem_', '_gt_', '_hash_',
'_iadd_', '_imul_', '_init_', '_init_subclass_', '_iter_', '_le_', '_len_', '_lt_',
'_mul_', '_ne_', '_new_', '_reduce_', '_reduce_ex_', '_repr_', '_reversed_',
'_rmul_', '_setattr_', '_setitem_', '_sizeof_', '_str_', '_subclasshook_', 'append',
'clear', 'copy', 'count', 'extend', 'index', 'insert', 'pop', 'remove', 'reverse', 'sort']
```

מתוך רשימה הזאת פונקציות הן:

sort, clear, copy, count, extend, index, insert, pop, remove, reverse, append,

sort

פונקציה מקבלת רשימה ומשנה את הסדר של האיברים מהקטן לגדול (לא עובדת אם במחרוזת יש תו או מחרוזת).

```
>>>a=[1,2,3,4,5,6,7,2,3,2,1]
>>> a
[1, 3, 4, 5, 6, 7, 3, 2, 1]
>>> a.sort()
>>> a
[1, 1, 2, 3, 3, 4, 5, 6, 7]
```

reverse

פונקציה מקבלת רשימה ומשנה את הסדר לסדר הפוך.

```
>>> a=[9,3,6,8,3,2,4,1,0,2.2,1.9]
>>> a
[9, 3, 6, 8, 3, 2, 4, 1, 0, 2.2, 1.9]
>>> a.reverse()
>>> a
[1.9, 2.2, 0, 1, 4, 2, 3, 8, 6, 3, 9]
```

remove

פונקציה מורידה מהרשימה איבר שאתם רוצים ע"י ציון הערך ולא ע"י ציון המיקום של הערך ברשימה. במידה שנמצא אותו ערך מספר פעמים, הפונקציה מורידה את הערך עם האינדקס הקטן ביותר.

```
>>> a
[1.9, 2.2, 0, 1, 4, 2, 3, 8, 6, 3, 9]
>>> a.remove(3)
>>> a
[1.9, 2.2, 0, 1, 4, 2, 8, 6, 3, 9]
>>> a.remove(3)
>>> a
[1.9, 2.2, 0, 1, 4, 2, 8, 6, 9]
>>> a.remove(3)
Traceback (most recent call last):
  File "<pyshell#29>", line 1, in <module>
    a.remove(3)
ValueError: list.remove(x): x not in list
```

pop פונקציה להוצאת אלמנט מהסוף של רשימה.

```
>>> a
[1.9, 2.2, 0, 1, 4, 2, 8, 6, 9]
>>> a.pop()
9
>>> a
[1.9, 2.2, 0, 1, 4, 2, 8, 6]
```

append פונקציה שמוסיפה אלמנט לסוף של רשימה.

```
>>> a
[1.9, 2.2, 0, 1, 4, 2, 8, 6]
>>> a.append(2.1)
>>> a
[1.9, 2.2, 0, 1, 4, 2, 8, 6, 2.1]
```

insert פונקציה שמקבלת 2 פרמטרים: פרמטר ראשון הוא האינדקס למיקום בו מכניסים איבר ופרמטר שני זה הערך עצמו שמכניסים לאיבר במיקום הנבחר.

```
>>> a
[1.9, 2.2, 0, 1, 4, 2, 8, 6, 2.1]
>>> a.insert(2,100)
>>> a
[1.9, 2.2, 100, 0, 1, 4, 2, 8, 6, 2.1]
```

clear פונקציה שמנקה את הרשימה מכל ערך קיים.

```
>>> a
[1.9, 2.2, 100, 0, 1, 4, 2, 8, 6, 2.1]
>>> a.clear()
>>> a
[]
```

פונקציה מחפשת ערך ברשימה ומחזירה את המיקום שלו. במידה ואותו ערך מופיע מספר פעמים, הפונקציה מחזירה את המיקום הראשון שבו הוא מופיע.

index

```
>>> a
[9, 3, 6, 8, 3, 2, 4, 1, 0, 2.2, 1.9]
>>> a.index(2.2)
9
>>> a.index(3)
1
```

פונקציה שמקבלת ערך ומחזירה את מספר פעמים שאיבר הזה מופיע ברשימה.

count

```
>>> a
[9, 3, 6, 8, 3, 2, 4, 1, 0, 2.2, 1.9]
>>> a.count(3)
2
```

פונקציה שמקבלת רשימה, ומצרפת אותה לסוף של רשימה אחרת נבחרת, שנחשבת לרשימת המקור. רשימת המקור נשארת ללא שינוי.

extend

```
>>> b=[1,2,3,4]
>>> a.extend(b)
>>> a
[9, 3, 6, 8, 3, 2, 4, 1, 0, 2.2, 1.9, 1, 2, 3, 4]
```

פונקציה שמעתיקה רשימת מקור לרשימת היעד.

copy

```
>>> b
[1, 2, 3, 4]
>>> a=b.copy()
>>> a
[1, 2, 3, 4]
```

אם שכחתם כיצד משתמשים בפונקציה מסוימת ניתן לקבל עזרה:

```
>>> help(list.index)
Help on method_descriptor:

index(self, value, start=0, stop=2147483647, /)
    Return first index of value.

    Raises ValueError if the value is not present.

>>> help(list.count)
Help on method_descriptor:

count(self, value, /)
    Return number of occurrences of value.
```

תרגיל דוגמה:

כתוב תוכנית שמייצרת רשימה ריקה וממלאת אותה ב-10 ערכים אקראיים בין 0 ל-100.

```
>>> import random
>>> c=[]
>>> c
[]
>>> i=0
>>> while i<10:
    c.append(random.randint(0,101))
    i+=1

>>> c
[25, 31, 60, 8, 87, 32, 39, 88, 76, 18]
```

מילוי ערכים לתוך רשימה.

מילוי ערכים אוטומטי בעזרת לולאת for.

```
>>> for i in range(10):
    a.append(i+1)
>>> a
[1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
```

בפייתון יש אפשרות לשלב שורות ולבצע משימה זו בצורה מהירה יותר:

```
>>> a=[i+1 for i in range(10)]
>>> a
[1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
```

מילוי ערכים אקראיים מ-1 עד 50.

```
>>> import random
>>> a=[]
>>> for i in range(10):
    a.append(random.randint(1,50))

>>> a
[48, 16, 1, 25, 9, 39, 2, 24, 49, 37]
```

עוד מילוי מעניין בעזרת for כפול (לא for בתוך for).

```
>>> s1='abcd'
>>> s2='01'
>>> [i+j for i in s1 for j in s2]
['a0', 'a1', 'b0', 'b1', 'c0', 'c1', 'd0', 'd1']
```

דוגמה עם for ו-if.

```
>>> a
[1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
>>> [i for i in a if i%2==0]
[2, 4, 6, 8, 10]
```

## דוגמאות בנושא רשימות (רמה מתקדמת):

דוגמה מס' 1.

כתוב תוכנית שסופרת ערכים מופיעים רק פעם אחת בתוך הרשימה.

(פתרון 1 – ללא פקודות מיוחדות של פיתון).

```
# taking an input list
input_list = [1, 2, 2, 5, 8, 4, 4, 8]
# create empty list
l1 = []

count = 0

# traversing the array
for item in input_list:
    if item not in l1:
        count += 1
        l1.append(item)

# printing the output
print("No of unique items are:", count)
```



(פתרון 2 – עם פקודה מיוחדת).

```
from collections import Counter
input_list = [1, 2, 2, 5, 8, 4, 4, 8]
# creating a list with the keys
items = Counter(input_list).keys()
print("No of unique items in the list are:", len(items))
```

דוגמה מס' 2.

תוכנית להצגת איברים ייחודיים ומכפלתם.

פתרון ראשון:

```
test_list = [1, 3, 5, 6, 3, 5, 6, 1]
print ("The original list is : " + str(test_list))
# using naive method
# Duplication Removal List Product
res = []
for i in test_list:
    if i not in res:
        res.append(i)
print(res)
res = prod(res)
# printing list after removal
print ("Duplication removal list product : " + str(res))
```

פתרון שני (בעזרת פונקציה):

```
def prod(val):
    res = 1
    for ele in val:
        res *= ele
    return res
# initializing list
test_list = [1, 3, 5, 6, 3, 5, 6, 1]
print("The original list is : " + str(test_list))
# using list comprehension
# Duplication Removal List Product
res = []
[res.append(x) for x in test_list if x not in res]
print(res)
res = prod(res)
# printing list after removal
print("Duplication removal list product : " + str(res))
```

דוגמה מס' 3.

מצא איברים של רשימה שמופיעים יותר ממספר מסוים מוגדר מראש או נקלט מהמשתמש.  
פתרון ראשון:

```
# initializing list
test_list = [4, 6, 4, 3, 3, 4, 3, 7, 8, 8]
# printing string
print("The original list : " + str(test_list))
# initializing K
K = 2
res = []
for i in test_list:
    # using count() to get count of elements
    freq = test_list.count(i)
    # checking if not already entered in results
    if freq > K and i not in res:
        res.append(i)
print("K=", K)
# printing results
print("The required elements : " + str(res))
```

פתרון שני:

```
from collections import Counter
# initializing list
test_list = [4, 6, 4, 3, 3, 4, 3, 7, 8, 8]

# printing string
print("The original list : " + str(test_list))

# initializing K
K = 2

# using list comprehension to bind result
res = [ele for ele, cnt in Counter(test_list).items() if cnt > K]
print("K= ",K)
# printing results
print("The required elements : " + str(res))
```

דוגמה מס' 4.

כתוב תוכנית שבודקת האם רשימה מכילה איברים בטווח הנתון.

שימוש בלולאות:

```
test_list = [4, 5, 6, 7, 3, 9]
# printing original list
print("The original list is : " + str(test_list))
# Initialization of range
i, j = 3, 10
# Test if List contains elements in Range
# using loop
res = True
for ele in test_list:
    if ele < i or ele >= j:
        res = False
        break
# printing result
print("Does list contain all elements in range : " + str(res))
```

שימוש בפונקציה :all()

```
test_list = [4, 5, 6, 7, 3, 9]
# printing original list
print("The original list is : " + str(test_list))
# Initialization of range
i, j = 3, 10
# Test if List contains elements in Range
# using all()
res = all(ele >= i and ele < j for ele in test_list)
# printing result
print("Does list contain all elements in range : " + str(res))
```

דוגמה מס' 5.

כתוב תוכנית שמחפשת איברים שמופיעים 3 פעמים ברצף בתוך רשימה:

```
# creating the array
arr = [4, 8, 5, 5, 5, 5, 5, 3, 3, 3, 3, 8, 8]

# size of the list
size = len(arr)

# looping till length - 2
for i in range(size - 2):

    # checking the conditions
    if arr[i] == arr[i + 1] and arr[i + 1] == arr[i + 2]:
        # printing the element as the
        # conditions are satisfied
        print(arr[i])
```

דוגמה מס' 6.

ניקח את הרשימה של מספרים, מכל זוג ניקח את מספר הגדול ונציג אותו בשורה חדשה. לדוגמה:

Input: 1 2 2 3 4 5

Output: 2 2 3 4 5

```
def maximumAdjacent(arr1, n):
    # array to store the max
    # value between adjacent pairs
    arr2 = []
    # iterate from 1 to n - 1
    for i in range(1, n):
        # find max value between
        # adjacent pairs gets
        # stored in r
        r = max(arr1[i], arr1[i - 1])
        # add element
        arr2.append(r)
    # printing the elements
    for ele in arr2:
        print(ele, end=" ")
if __name__ == "__main__":
    # size of the input array
    n = 6
    # input array
```

```
arr1 = [1, 2, 2, 3, 4, 5]
# function calling
maximumAdjacent(arr1, n)
```

דוגמה מס' 7.

נכתוב תוכנית שמקבלת 3 ספרות ומדפיסה את כל הסידורים של ספרות האלה. לדוגמה:

Input: [1, 2, 3]

Output:

```
1 2 3
1 3 2
2 1 3
2 3 1
3 1 2
3 2 1
```

```
def comb(L):
    for i in range(3):
        for j in range(3):
            for k in range(3):
                # check if the indexes are not
                # same
                if (i!=j and j!=k and i!=k):
                    print(L[i], L[j], L[k])
# Driver Code
comb([1, 2, 3])
```

דוגמה מס' 8.

נכתוב תוכנית שמקבלת רשימה של רשימות ומדפיסה את הסידורים של רשימות האלה.

```
# initializing list
test_list = [["http", "www", "elecstudy", "org"], [5, 4, 3, 4], "is",
             ["best", "good", "better", "average"]]

# printing original list
print("The original list is : " + str(test_list))
```

```

# initializing size of inner Optional list
K = 4

res = []
cnt = 0
while cnt <= K - 1:
    temp = []

    # inner elements selections
    for idx in test_list:

        # checks for type of Elements
        if not isinstance(idx, list):
            temp.append(idx)
        else:
            temp.append(idx[cnt])
    cnt += 1
    res.append(temp)

# printing result
print("All index Combinations : " + str(res))

```

דוגמה מס' 9.

נכתוב תוכנית שמקבלת רשימה של רשימות ומדפיסה את כל הסידורים של רשימות האלה.

```

List_1 = ["a","b"]
List_2 = [1,2]
combination = [[('a',1),('b',2)],[('a',2),('b',1)]]

import itertools
from itertools import permutations

# initialize lists
list_1 = ["a", "b", "c", "d"]
list_2 = [1,4,9]

# create empty list to store the
# combinations
unique_combinations = []

# Getting all permutations of list_1
# with length of list_2
permut = itertools.permutations(list_1, len(list_2))

```

```

# zip() is called to pair each permutation
# and shorter list element into combination
for comb in permut:
    zipped = zip(comb, list_2)
    unique_combinations.append(list(zipped))

# printing unique_combination list
print(unique_combinations)

```

דוגמה מס' 10.

נכתוב תוכנית שמקבלת רשימה ומספר ומדפיסה את הרשימה לא כולל מספר זה.

Input :

1 1 2 3 4 5 1 2

1

Output :

2 3 4 5 2

פתרון ראשון:

```

def remove_items(test_list, item):
    # using list comprehension to perform the task
    res = [i for i in test_list if i != item]
    return res

# driver code
if __name__ == "__main__":

    # initializing the list
    test_list = [1, 3, 4, 6, 5, 1]

    # the item which is to be removed
    item = 1

    # printing the original list
    print ("The original list is : " + str(test_list))

    # calling the function remove_items()
    res = remove_items(test_list, item)

    # printing result
    print ("The list after performing the remove operation is : " + str(res))

```

```

def remove_items(test_list, item):
    # using filter() + __ne__ to perform the task
    res = list(filter((item).__ne__, test_list))

    return res

# driver code
if __name__ == "__main__":

    # initializing the list
    test_list = [1, 3, 4, 6, 5, 1]

    # the item which is to be removed
    item = 1

    # printing the original list
    print ("The original list is : " + str(test_list))

    # calling the function remove_items()
    res = remove_items(test_list, item)

    # printing result
    print ("The list after performing the remove operation is : " + str(res))

```

```

def remove_items(test_list, item):
    # remove the item for all its occurrences
    for i in test_list:
        if(i == item):
            test_list.remove(i)
    return test_list

# driver code
if __name__ == "__main__":

    # initializing the list
    test_list = [1, 3, 4, 6, 5, 1]

    # the item which is to be removed
    item = 1

    # printing the original list

```



```

print ("The original list is : " + str(test_list))

# calling the function remove_items()
res = remove_items(test_list, item)

# printing result
print ("The list after performing the remove operation is : " + str(res))

```

תרגול בנושא רשימות:  
תרגול בנושא טבלאות מעקב.  
תרגיל מס' 1.

תפענח את הקוד הבא. מהו פלט אפשרי של בכל מקרה של קלט.

```

import random
def RandNum():
    l = []
    x1,x2 = random.sample(range(-10, 10), 2)
    N = random.randrange(1,4)
    if N == 1:
        l.append(x1)
        l.append(x1)
        l.append(x2)
    elif N == 2:
        l.append(x1)
        l.append(x2)
        l.append(x2)
    else:
        l.append(x2)
        l.append(x1)
        l.append(x2)
    return l
A,B,C = RandNum()
print("Number A:", A)
print("Number B:", B)
print("Number C:", C)
if A == B:
    print("C")
elif A == C:
    print("B")
else:
    print("A")

```

תפענח את הקוד הבא. מהו פלט אפשרי של בכל מקרה של קלט.

```
import random
def RandNum():
    l = []
    x1,x2 = random.sample(range(-10, 10), 2)
    N = random.randrange(1,5)
    if N == 1:
        l.append(x2)
        l.append(x1)
        l.append(x1)
        l.append(x1)
    elif N == 2:
        l.append(x1)
        l.append(x2)
        l.append(x1)
        l.append(x1)
    elif N == 3:
        l.append(x1)
        l.append(x1)
        l.append(x2)
        l.append(x1)
    else:
        l.append(x1)
        l.append(x1)
        l.append(x1)
        l.append(x2)
    return l
A,B,C,D = RandNum()
print("Number A:", A)
print("Number B:", B)
print("Number C:", C)
print("Number D:", D)
if A == B:
    if A == D:
        print("C")
    else:
        print("D")
else:
    if A == C:
        print("B")
    else:
        print("A")
```

תפענח את הקוד הבא. מהו פלט אפשרי של בכל מקרה של קלט.

```
import random
lst = [i for i in list(range(-10,11)) if i != 0]
for i in range(0,5):
    x = random.choice(lst)
    y = random.choice(lst)
    print("\nТочка (x, y): ({0},{1})".format(x, y))
    print("Quarter: ", end="")
    if x > 0 and y > 0:
        print("I")
    elif x < 0 and y > 0:
        print("II")
    elif x < 0 and y < 0:
        print("III")
    else:
        print("IV")
```

תרגיל מס' 4.

תפענח את הקוד הבא. מהו פלט אפשרי של בכל מקרה של קלט.

```
import random
x1,x2 = sorted(random.sample(range(-10, 11), 2))
y2,y1 = sorted(random.sample(range(-10, 11), 2))
L = [[x1,y1], [x1,y2], [x2,y1], [x2,y2]]
print(L)
random.shuffle(L)
print(L)
M = L[:3]
print(M)
print("Apex 1: ({0},{1})".format(M[0][0], M[0][1]))
print("Apex 2: ({0},{1})".format(M[1][0], M[1][1]))
print("Apex 3: ({0},{1})".format(M[2][0], M[2][1]))
print()
min_x = min(M[0][0],M[1][0],M[2][0])
min_y = min(M[0][1],M[1][1],M[2][1])
max_x = max(M[0][0],M[1][0],M[2][0])
max_y = max(M[0][1],M[1][1],M[2][1])
L = [[min_x,min_y], [min_x,max_y], [max_x,min_y], [max_x,max_y]]
for i in L:
    if i not in M:
        print("Apex 4: ({0},{1})".format(i[0], i[1]))
```

תרגיל מס' 5.

תפענח את הקוד הבא. מהו פלט אפשרי של בכל מקרה של קלט.

```
L = [1,21,80,99,100,101,800,901,999]
for i in L:
    if i%2 == 0:
        s = "even "
    else:
        s = "odd "
    i_str = str(i)
    n = len(i_str)
    if n == 1:
        s += "one digit "
    elif n == 2:
        s += "two digits "
    elif n == 3:
        s += "three digits "
    s += "Number"
    print(i, " : ",s)
```

תרגיל מס' 6.

תפענח את הקוד הבא. מהו פלט אפשרי של בכל מקרה של קלט.

```
L = [2016,300,1300,1900,1200,2000]
for i in L:
    s = "Not Leap"
    if (i%4 == 0) and not(i%100 == 0 and i%400 != 0):
        s = "Leap"
    print(i, " : ",s)

    if (i%4 == 0) and (i%100!= 0 or i%400 == 0):
        s = "Leap"
    print(i, " : ",s)
```

תרגיל מס' 7.

תפענח את הקוד הבא. מהו פלט אפשרי של בכל מקרה של קלט.

```
import random
fib = []
def Fib1(N):
    if N < len(fib):
        return fib[N-1]
    if N == 1 or N == 2:
```

```

    if N > len(fib):
        fib.append(1)
    return 1
y = Fib1(N-2) + Fib1(N-1)
if N > len(fib):
    fib.append(y)
return y
K = random.randint(1,40)
print("K = ",K)
N = Fib1(K)
print("Fibonacci Number (N): ", N)
F1 = F2 = 1
print(1,":",F1)
print(2,":",F2)
i = 2
while F1 < N:
    F0, F1, F2 = F1, F2, F1+F2
    i += 1
    print(i,":",F1)
print()
print("{0} + {1} = {2}".format(F0,N,F2))

```

תרגיל מס' 8.

תפענח את הקוד הבא. מהו פלט אפשרי של בכל מקרה של קלט.

```

from random import randint

def search(lst, item):
    mid = len(lst) // 2
    low = 0
    high = len(lst) - 1
    while lst[mid] != item and low <= high:
        if item > lst[mid]:
            low = mid + 1
        else:
            high = mid - 1
        mid = (low + high) // 2
    if low > high:
        return None
    else:
        return mid
a = []
for i in range(10):
    a.append(randint(1, 20))

```

```
a.sort()
print(a)
value = int(input())
print(search(a, value))
```

תרגיל מס' 9.

תפענח את הקוד הבא. מהו פלט אפשרי של בכל מקרה של קלט.

```
num_list = [1, -5, 3, 8, 0, 9, -6, 3, -1, 5]
print(num_list)
for i in range(len(num_list)):
    if num_list[i] > 0:
        num_list[i] = 1
    elif num_list[i] < 0:
        num_list[i] = -1
print(num_list)
```

תרגיל מס' 10.

תפענח את הקוד הבא. מהו פלט אפשרי של בכל מקרה של קלט.

```
from random import randint
def fill_list(lst, qty, low, high):
    for i in range(qty):
        lst.append(randint(low, high))

minimum = int(input("Min: "))
maximum = int(input("Max: "))

n = int(input("Quantity: "))
a = []

fill_list(a, n, minimum, maximum)
print(a)
```

תרגיל מס' 11.

תפענח את הקוד הבא. מהו פלט אפשרי של בכל מקרה של קלט.

```
num_list = [1, -5, 3, 8, 0, 9, -6, 3, -1, 5]
print(num_list)
for index, value in enumerate(num_list):
    if value > 0:
        num_list[index] = 1
```

```
elif value < 0:  
    num_list[index] = -1  
print(num_list)
```

תרגיל מס' 12.

תפענח את הקוד הבא. מהו פלט אפשרי של בכל מקרה של קלט.

```
import random  
a = []  
for i in range(20):  
    n = round(random.random() * 100)  
    a.append(n)  
print("A =", a)  
b = []  
i = 0  
while i < len(a):  
    if 35 < a[i] < 65:  
        b.append(a[i])  
        del a[i]  
    else:  
        i += 1  
print("A =", a)  
print("B =", b)
```

תרגיל מס' 13.

תפענח את הקוד הבא. מהו פלט אפשרי של בכל מקרה של קלט.

```
import random  
fib = []  
def Fib1(N):  
    if N < len(fib):  
        return fib[N-1]  
    if N == 1 or N == 2:  
        if N > len(fib):  
            fib.append(1)  
        return 1  
    y = Fib1(N-2) + Fib1(N-1)  
    if N > len(fib):  
        fib.append(y)  
    return y  
I = random.randint(1,40)  
print("I =", I)  
N = Fib1(I)
```

```

print("Fibonacci Number (N): ", N)
F1 = F2 = 1
i = 2
while F2 < N:
    F0, F1, F2 = F1, F2, F1+F2
    i += 1
print()
print(i,":",F2)

```

תרגיל מס' 14.

תפענח את הקוד הבא. מהו פלט אפשרי של בכל מקרה של קלט.

```

num_list = [1, -5, 3, 8, 0, 9, -6, 3, -1, 5]
print(num_list)
neg = pos = zero = 0
for i in num_list:
    if i > 0:
        pos += 1
    elif i < 0:
        neg += 1
    else:
        zero += 1
print("Positive: ", pos)
print("Negative: ", neg)
print("Zero: ", zero)

```

תרגיל מס' 15.

תפענח את הקוד הבא. מהו פלט אפשרי של בכל מקרה של קלט.

```

def shift(lst, steps):
    if steps < 0:
        steps = abs(steps)
        for i in range(steps):
            lst.append(lst.pop(0))
    else:
        for i in range(steps):
            lst.insert(0, lst.pop())

nums = [4, 5, 6, 7, 8, 9, 0]
print(nums)

shift(nums, -2)

```



```
print(nums)
shift(nums, 3)
print(nums)
```

תרגיל מס' 16.

תפענח את הקוד הבא. מהו פלט אפשרי של בכל מקרה של קלט.

```
def fill_list(m1, m2, amount, l):
    from random import randint
    for i in range(amount):
        l.append(randint(m1, m2))
def analysis(your_list, your_dict):
    for i in your_list:
        if i in your_dict:
            your_dict[i] += 1
        else:
            your_dict[i] = 1
lst = []
dct = {}
mn = int(input('Minimum: '))
mx = int(input('Maximum: '))
qty = int(input('Number of elements: '))
fill_list(mn, mx, qty, lst)
analysis(lst, dct)
print(lst)
for item in sorted(dct):
    print("%d:%d" % (item, dct[item]),end=" ")
```

תרגיל מס' 17.

תפענח את הקוד הבא. מהו פלט אפשרי של בכל מקרה של קלט.

```
from random import randint
pos = []
neg = []
for i in range(20):
    n = randint(-5, 5)
    if n > 0:
        pos.append(n)
    elif n < 0:
        neg.append(n)
print(pos)
print(neg)
```

תרגיל מס' 18.

תפענח את הקוד הבא. מהו פלט אפשרי של בכל מקרה של קלט.

```
from random import randint
num_list = [randint(1, 99) for i in range(10)]
print(num_list)
limit = int(input("Limit is: "))
for index, value in enumerate(num_list):
    if value < limit:
        num_list[index] = limit
print(num_list)
```

תרגיל מס' 19.

תפענח את הקוד הבא. מהו פלט אפשרי של בכל מקרה של קלט.

```
from random import randint
N = 10
arr = []
for i in range(N):
    arr.append(randint(-50, 49))
print(arr)
num = 0
for i in range(1, N):
    if abs(arr[i]) < abs(arr[num]):
        num = i
print("The minimal element is:")
print("arr[{}] = {}".format(num, arr[num]))
```

תרגיל מס' 20.

תפענח את הקוד הבא. מהו פלט אפשרי של בכל מקרה של קלט.

```
from random import randint
N = 10
a = [randint(0, 30) for i in range(N)]
print(a)
max1_index = 0
for i in range(1, N):
    if a[i] > a[max1_index]:
        max1_index = i
max2_index = 0
if max2_index == max1_index:
    max2_index = 1
```

```

for i in range(1, N):
    if i == max1_index:
        continue
    if a[i] > a[max2_index]:
        max2_index = i
print("First Maximum is =", a[max1_index])
print("Second Maximum is =", a[max2_index])

```

תרגיל מס' 21.

תפענח את הקוד הבא. מהו פלט אפשרי של בכל מקרה של קלט.

```

from random import randint
N = 10
a = []
suma = 0
for i in range(N):
    a.append(randint(0, 9))
    print(a[i], end=' ')
    suma += a[i]
print()
average = suma / N
print("Average is: %.2f" % average)
for i in a:
    if i > average:
        print(i, end=' ')
print()

```

תרגיל מס' 22.

תפענח את הקוד הבא. מהו פלט אפשרי של בכל מקרה של קלט.

```

from random import randint
N = 15
a = [randint(-10, 10) for i in range(N)]
print(a)
j = 1
v1 = v3 = v6 = 0
for value in a:
    if value > 0:
        if j == 1:
            v1 = value
        elif j == 3:
            v3 = value
        elif j == 6:

```

```

v6 = value
break
j += 1
mult = v1 * v3 * v6
print("%d * %d * %d = %d" % (v1, v3, v6, mult))

```

תרגיל מס' 23.

תפענח את הקוד הבא. מהו פלט אפשרי של בכל מקרה של קלט.

```

a = [5, [1, 2], 2, 'r', 4, 'ee']
b = [4, 'we', 'ee', 3, [1, 2]]
c = []
for i in a:
    for j in b:
        if i == j:
            c.append(i)
            break
print(c)

```

תרגיל מס' 24.

תפענח את הקוד הבא. מהו פלט אפשרי של בכל מקרה של קלט.

```

from random import random
N = 10
a = [0]*N
for i in range(N):
    a[i] = int(random()*50)
    print('%3d' % a[i], end="")
print()
min_id = 0
max_id = 0
for i in range(1, N):
    if a[i] < a[min_id]:
        min_id = i
    elif a[i] > a[max_id]:
        max_id = i
print(a[min_id], a[max_id])
if min_id > max_id:
    min_id, max_id = max_id, min_id
summa = 0
for i in range(min_id+1, max_id):
    summa += a[i]
print(summa)

```

תרגיל מס' 25.

תפענח את הקוד הבא. מהו פלט אפשרי של בכל מקרה של קלט.

```
from random import randint
N = 10
a = [randint(1, 50) for i in range(N)]
print(a)
qty_big=0
i = 1
if a[i-1] > a[i]:
    qty_big += 1
    i = 2
if a[N-1] > a[N-2]:
    qty_big += 1
    N -= 1
while i < N-1:
    if a[i-1] < a[i] > a[i+1]:
        qty_big += 1
        i += 2
    else:
        i += 1
print(qty_big)
```

תרגיל מס' 26.

תפענח את הקוד הבא. מהו פלט אפשרי של בכל מקרה של קלט.

```
a = [3, 5, 7, 3, 8, 1, 8, 1, 7, 3, 2,
4, 6, 8, 5, 4, 3, 3, 6, 5, 7, 8,
9, 5, 3, 2, 3]
count_1_3 = 0
count_4_6 = 0
count_7_9 = 0
for i in a:
    if 1 <= i <= 3:
        count_1_3 += 1
    elif 4 <= i <= 6:
        count_4_6 += 1
    elif 7 <= i <= 9:
        count_7_9 += 1
print("Range 1-3:", count_1_3, "items")
print("Range 4-6:", count_4_6, "items")
print("Range 7-9:", count_7_9, "items")
```

תרגיל מס' 27.

תפענח את הקוד הבא. מהו פלט אפשרי של בכל מקרה של קלט.

```
from random import random
N = 10
arr = [0] * N
for i in range(N):
    arr[i] = int(random() * 50)
print(arr)

for i in range(N-1):
    for j in range(i+1, N):
        if arr[i] == arr[j]:
            print("There are some equal elements.")
            quit()
print("All elements are unique")
```

תרגיל מס' 28.

תפענח את הקוד הבא. מהו פלט אפשרי של בכל מקרה של קלט.

```
from random import randint
N = 15
a = [randint(-10, 10) for i in range(N)]
print(a)
b = list(filter(lambda x: x > 0, a))
mult = b[0] * b[2] * b[5]
print("%d * %d * %d = %d" % (b[0], b[2], b[5], mult))
```

תרגיל מס' 29.

תפענח את הקוד הבא. מהו פלט אפשרי של בכל מקרה של קלט.

```
from random import randint
N = 10
a = [randint(0, 30) for i in range(N)]
print(a)
b = sorted(a)
max1_value = b[-1]
max2_value = b[-2]
max1_index = a.index(max1_value)
max2_index = a.index(max2_value)
print("First maximum is =", a[max1_index])
print("Second maximum is =", a[max2_index])
```

תרגיל מס' 30.

תפענח את הקוד הבא. מהו פלט אפשרי של בכל מקרה של קלט.

```
from random import randint
N = 10
arr = [randint(-50, 49) for i in range(N)]
print(arr)
arr2 = [abs(i) for i in arr]
index_min = arr2.index(min(arr2))
print("Minimal Element is:")
print("arr[{}] = {}".format(index_min, arr[index_min]))
```

תרגיל מס' 31.

תפענח את הקוד הבא. מהו פלט אפשרי של בכל מקרה של קלט.

```
from random import randint
origin_list = [randint(-5, 5) for i in range(20)]
pos_list = list(filter(lambda x: x > 0, origin_list))
neg_list = list(filter(lambda x: x < 0, origin_list))
print(pos_list)
print(neg_list)
```

תרגיל מס' 32.

תפענח את הקוד הבא. מהו פלט אפשרי של בכל מקרה של קלט.

```
a = [5, [1, 2], 2, 'r', 4, 'ee', 'ee']
b = [4, 'we', 'ee', 3, [1, 2]]
c = []
for i in a:
    if i in c:
        continue
    for j in b:
        if i == j:
            c.append(i)
            break
print(c)
```

תרגיל מס' 33.

תפענח את הקוד הבא. מהו פלט אפשרי של בכל מקרה של קלט.

```
from random import randint
N = 10
a = [randint(1, 50) for i in range(N)]
print(a)
qty_big = 0
for i in range(1, N-1):
    if a[i-1] < a[i] > a[i+1]:
        qty_big += 1
if a[0] > a[1]:
    qty_big += 1
if a[N-1] > a[N-2]:
    qty_big += 1
print(qty_big)
```

תרגיל מס' 34.

תפענח את הקוד הבא. מהו פלט אפשרי של בכל מקרה של קלט.

```
a = [3, 5, 7, 3, 8, 1, 8, 1, 7, 3, 2,
4, 6, 8, 5, 4, 3, 3, 6, 5, 7, 8,
9, 5, 3, 2, 3]

count_1_3 = len(list(filter(lambda i: 1 <= i <= 3, a)))
count_4_6 = len(list(filter(lambda i: 4 <= i <= 6, a)))
count_7_9 = len(list(filter(lambda i: 7 <= i <= 9, a)))
print("Range 1-3:", count_1_3, "items")
print("Range 4-6:", count_4_6, "items")
print("Range 7-9:", count_7_9, "items")
```

תרגיל מס' 35.

תפענח את הקוד הבא. מהו פלט אפשרי של בכל מקרה של קלט.

```
a = [5, 2, 'r', 4, 'ee']
b = [4, 1, 'we', 'ee', 2, 'r']
c = list(set(a) & set(b))
print(c)
c = list(set(a) | set(b))
print(c)
c = list(set(b) - set(a))
print(c)
```



```
c = list(set(a) ^ set(b))
print(c)
```

תרגיל מס' 36.

תפענח את הקוד הבא. מהו פלט אפשרי של בכל מקרה של קלט.

```
from random import random
N = 10
arr = [0] * N
for i in range(N):
    arr[i] = int(random() * 50)
print(arr)

setarr = set(arr)
if len(arr) == len(setarr):
    print("There are some equal elements.")
else:
    print("All elements are unique")
```

תרגיל מס' 37.

תפענח את הקוד הבא. מהו פלט אפשרי של בכל מקרה של קלט.

```
from random import randint
N = 10
a = [randint(10, 20) for i in range(N)]
print(a)
unic_set = set(a)
for i in unic_set:
    if a.count(i) == 1:
        print(i, end=" ")
```

תרגיל מס' 38.

תפענח את הקוד הבא. מהו פלט אפשרי של בכל מקרה של קלט.

```
from random import random
N = 15
arr = [0] * N
for i in range(N):
    arr[i] = int(random() * 20)
print(arr)
num = arr[0]
max_frq = 1
```

```

for i in range(N-1):
    frq = 1
    for k in range(i+1,N):
        if arr[i] == arr[k]:
            frq += 1
    if frq > max_frq:
        max_frq = frq
        num = arr[i]
if max_frq > 1:
    print(max_frq, 'times, the number', num)
else:
    print('All elements are unique')

```

תרגיל מס' 39.

תפענח את הקוד הבא. מהו פלט אפשרי של בכל מקרה של קלט.

```

from random import random
N = 10
a = [0]*N
for i in range(N):
    a[i] = int(random()*50)
    print('%3d' % a[i], end="")
print()
min_id = a.index(min(a))
max_id = a.index(max(a))
print(a[min_id], a[max_id])
if min_id > max_id:
    min_id, max_id = max_id, min_id
summa = sum(a[min_id+1:max_id])
print(summa)

```

תרגיל מס' 40.

תפענח את הקוד הבא. מהו פלט אפשרי של בכל מקרה של קלט.

```

a = [5, 2, 'r', 4, 'ee']
b = [4, 1, 'we', 'ee', 2, 'r']
c = list(set(a) & set(b))
print(c)

```

תרגיל מס' 41.

תפענח את הקוד הבא. מהו פלט אפשרי של בכל מקרה של קלט.

```
from random import random
N = 20
a = [0] * N
for i in range(N):
    a[i] = int(random()*15)
    print(a[i],end=' ')
print()
for i in range(N):
    f = True
    for j in range(N):
        if a[i] == a[j] and i != j:
            f = False
            break
    if f == True:
        print(a[i],end=' ')
print()
```

תרגיל מס' 42.

תפענח את הקוד הבא. מהו פלט אפשרי של בכל מקרה של קלט.

```
from random import randint
a = []
for i in range(10):
    a.append(randint(1, 50))
a.sort()
print(a)
value = int(input())
mid = len(a) // 2
low = 0
high = len(a) - 1
while a[mid] != value and low <= high:
    if value > a[mid]:
        low = mid + 1
    else:
        high = mid - 1
    mid = (low + high) // 2
if low > high:
    print("No value")
else:
    print("ID =", mid)
```

תרגיל מס' 43.

תפענח את הקוד הבא. מהו פלט אפשרי של בכל מקרה של קלט.

```
from random import randint
def sel_sort(array):
    for i in range(len(array) - 1):
        m = i
        j = i + 1
        while j < len(array):
            if array[j] < array[m]:
                m = j
            j = j + 1
        array[i], array[m] = array[m], array[i]
a = []
for i in range(10):
    a.append(randint(1, 99))
print(a)
sel_sort(a)
print(a)
```

תרגיל מס' 44.

תפענח את הקוד הבא. מהו פלט אפשרי של בכל מקרה של קלט.

```
from random import randint
def bubble(array):
    for i in range(N-1):
        for j in range(N-i-1):
            if array[j] > array[j+1]:
                buff = array[j]
                array[j] = array[j+1]
                array[j+1] = buff
N = 10
a = []
for i in range(N):
    a.append(randint(1, 99))
print(a)
bubble(a)
print(a)
```

תרגיל מס' 45.

תפענח את הקוד הבא. מהו פלט אפשרי של בכל מקרה של קלט.

```
def Repeat(x):
    _size = len(x)
    repeated = []
    for i in range(_size):
        k = i + 1
        for j in range(k, _size):
            if x[i] == x[j] and x[i] not in repeated:
                repeated.append(x[i])
    return repeated
list1 = [10, 20, 30, 20, 20, 30, 40, 50, -20, 60, 60, -20, -20]
print(Repeat(list1))
```

תרגיל מס' 46.

תפענח את הקוד הבא. מהו פלט אפשרי של בכל מקרה של קלט.

```
from random import randint
N = 10
arr = []
for i in range(N):
    arr.append(randint(1, 99))
print(arr)
i = 0
while i < N - 1:
    m = i
    j = i + 1
    while j < N:
        if arr[j] < arr[m]:
            m = j
        j += 1
    arr[i], arr[m] = arr[m], arr[i]
    i += 1
print(arr)
```

תרגיל מס' 47.

תפענח את הקוד הבא. מהו פלט אפשרי של בכל מקרה של קלט.

```
from random import randint
N = 10
a = []
```

```

for i in range(N):
    a.append(randint(1, 99))
print(a)
for i in range(N-1):
    for j in range(N-i-1):
        if a[j] > a[j+1]:
            a[j], a[j+1] = a[j+1], a[j]
print(a)

```

תרגיל מס' 48.

תפענח את הקוד הבא. מהו פלט אפשרי של בכל מקרה של קלט.

```

n = int(input())
a = []
for i in range(n + 1):
    a.append(i)
a[1] = 0
i = 2
while i <= n:
    if a[i] != 0:
        j = i + i
        while j <= n:
            a[j] = 0
            j = j + i
        i += 1
a = set(a)
a.remove(0)
print(a)

```

תרגיל מס' 49.

תפענח את הקוד הבא. מהו פלט אפשרי של בכל מקרה של קלט.

```

from random import randint
N = 10
a = []
for i in range(N):
    a.append(randint(1, 99))
print(a)
for i in range(N-1):
    flag=0
    for j in range(N-i-1):
        if a[j] > a[j+1]:
            flag=1

```

```
a[j], a[j+1] = a[j+1], a[j]
if(flag==0):
    break
print(a)
```

תרגיל מס' 50.

תפענח את הקוד הבא. מהו פלט אפשרי של בכל מקרה של קלט.

```
from random import randint
N = 10
a = []
for i in range(N):
    a.append(randint(1, 99))
print(a)
i = 0
while i < N - 1:
    j = 0
    while j < N - 1 - i:
        if a[j] > a[j+1]:
            a[j], a[j+1] = a[j+1], a[j]
        j += 1
    i += 1
print(a)
```

תרגיל מס' 51.

תפענח את הקוד הבא. מהו פלט אפשרי של בכל מקרה של קלט.

```
list1 = [11, 5, 17, 18, 23, 50, 3, 12, 11, 10]
for ele in list1:
    if ele % 2 == 0:
        list1.remove(ele)
print("New list after removing: ", list1)
```

תרגיל מס' 52.

תפענח את הקוד הבא. מהו פלט אפשרי של בכל מקרה של קלט.

```
list1 = [10, 20, 4, 45, 99, 34, 88, 23]
mx = max(list1[0], list1[1])
secondmax = min(list1[0], list1[1])
```

```

n = len(list1)
for i in range(2, n):
    if list1[i] > mx:
        secondmax = mx
        mx = list1[i]
    elif list1[i] > secondmax and mx != list1[i]:
        secondmax = list1[i]
print("The number is : ", str(secondmax))

```

תרגיל מס' 53.

תפענח את הקוד הבא. מהו פלט אפשרי של בכל מקרה של קלט.

```

def countX(lst, x):
    count = 0
    for ele in lst:
        if (ele == x):
            count = count + 1
    return count

lst = [8, 6, 8, 10, 8, 20, 10, 8, 8]
x = 8
print('{} has occurred {} times'.format(x, countX(lst, x)))

```



## תרגול בנושא כתיבת קוד.

### תרגיל מס' 1.

נתונה רשימה של מספרים שלמים. במקום מספרים שליליים שמים 1-, במקום חיוביים שמים 1, האפסים נשארים ללא שינוי.

### תרגיל מס' 2.

נתונה רשימה של מספרים שלמים. יש לחשב כמה מתוכם מספרים שליליים, כמה חיוביים וכמה אפסים.

### תרגיל מס' 3.

יש למלא רשימה במספרים אקראיים בין 1 לבין 100. מהרשימה יש למחוק איברים שנמצאים בתחום בין 35 עד 65 ויש לשמור אותם ברשימה אחרת.

בפיתון, ניתן להשתמש בפקודת del כדי להסיר פריט מהרשימה על ידי ציון הרשימה עצמה והאינדקס של הפריט שיש להסיר.

האלגוריתם לפתרון הבעיה נראה פשוט. מספיק למיין את פריטי הרשימה ולמחוק את אלה העומדים בתנאי. עם זאת, כאשר אלמנט נמחק, הבא נעמד למקומו, אך מכיוון שאנו עוברים לאלמנט הבא אנו מדלגים על בדיקה שהחלפנו אותו. לא ניתן להשתמש בלולאה for מכיוון שמספר פריטי הרשימה משתנה.

ניתן להשתמש בלולאת while, במדידת אורך הרשימה בכל איטרציה, תוך הגדלת האינדקס רק אם הפריט לא נמחק.

### תרגיל מס' 4.

התוכנית מייצרת רשימה של מספרים אקראיים בתחום בין 5- עד 5 (כולל קצוות). יש להעביר את כל המספרים החיוביים לרשימה X ואת כל המספרים השלילים לרשימה Y, אין לגעת באפסים. יש להציג רשימה X, רשימה Y ומה שנשאר מהרשימה המקורית.

### תרגיל מס' 5.

יש למלא רשימה של מספרים אקראיים ולהדפיס אותם על המסך. תקלוט מהמשתמש מספר כלשהו והמספר המינימלי שמשתמש רוצה להשאיר. כל המספרים מתחת למספר המינימלי יש להחליף במספר שנקלט מהמשתמש. יש להדפיס גם רשימה מקורית וגם רשימה לאחר שינוי.

תרגיל מס' 6.

יש למצוא את האיבר הקטן ביותר בערך מוחלט מתוך הרשימה. יש להדפיס את המספר עצמו ואת המיקום שלו.

תרגיל מס' 7.

ברשימה של מספרים שלמים צריכים למצוא 2 מספרים הגדולים ביותר ולהדפיס אותם על המסך.

תרגיל מס' 8.

יש להדפיס את כל האיברים של הרשימה שגדולים מהמוצע של כל המספרים.

תרגיל מס' 9.

ברשימה שמורכבת ממספרים שלמים יש למצוא מספרים חיוביים ולהעתיק אותם לרשימה חדשה. מרשימה החדשה יש להדפיס את המספר הראשון, המספר השלישי והמספר השישי ולהדפיס את מכפלתם.

תרגיל מס' 10.

יש למצוא חיתוך של 2 רשימות ולהדפיס אותו.

תרגיל מס' 11.

ברשימה חד ממדית מצא את סכום המספרים שנמצאים בין המספר הקטן ביותר לבין המספר הגדול ביותר לא כולל קצוות.

תרגיל מס' 12.

כתוב תוכנית שבודקת כמות איברים במערך שערכם יותר גדול מערכם של השכנים שלהם. לדוגמה 2 7 3 9 8 נוהצג 2 איברים.

כדי לענות על השאלה, יש להשוות בין כל אחד מהאברים ברשימה לבין הקרוב ולבין הבא. אם הוא גדול מהם, הגדל את המספר.

בפריט הראשון והאחרון יש רק "שכן" אחד. לכן אם הראשונים והאחרונים נלקחים בחשבון, יש לעבד אותם בנפרד.

תרגיל מס' 13.

נתונה רשימה עם מספרים מ-1 עד 9 כולל. כתוב תוכנית שבודקת כמה מספרים נמצאים בתחום בין 1-3, 4-6, 7-9.

לפעמים יש צורך לנתח את נתוני המדגם לפיזור הערכים על פני טווחים. התוצאה המתקבלת בדרך זו משמשת לבניית גרפים ותרשימים של תדרי התרחשות הערכים.

נתונה רשימת מספרים בין 1 ל 9 כולל. זה נדרש כדי לגלות כמה ערכים נפלו בטווח שבין 1 ל 3, בין 4 ל 6, בין 7 ל 9.

לשם כך אנו מציגים שלושה מונים תואמים. בשלב הבא, אנו נחלץ כל רכיב מהרשימה ונבדוק אם הוא נכנס לטווח כזה או אחר. במקרה זה, המונה של האלמנטים בטווח זה חייב להגדיל.

תרגיל מס' 14.

בדוק האם כל האיברים נמצאים רק פעם אחת בתוך הרשימה.

ישנן מספר דרכים לפתור את הבעיה בפיתון. האפשרות הקלאסית היא לקחת את מרכיבי הרשימה בתורם ולהשוות כל אחד לאלה שמאחוריו. בהתאמה הראשונה של האלמנטים ניתן להסיק כי הרשימה מכילה את אותם האלמנטים והתוכנית מסתיימת.

פתרון נוסף יכול להיות שימוש בסוג מערך הנתונים. כידוע, קבוצות אינן יכולות להכיל אלמנטים זהים. בעת המרת רשימה לקבוצה בה, אותם אלמנטים יוצגו פעם אחת, כלומר כפילויות יימחקו.

אם לאחר מכן נשווה את אורך הרשימה המקורית והתפאורה, ברור אם ברשימה יש אותם אלמנטים. אם האורכים זהים, אז כל האלמנטים ברשימה ייחודיים. אם לא, אז היו אותם אלמנטים וכבר אינם.

תרגיל מס' 15.

מצא את כל האיברים שנמצאים רק פעם אחת בתוך הרשימה ותדפיס אותם על המסך.

הדרך הקלה ביותר לפתור בעיה זו היא על ידי הפיכת הרשימה המקורית לסט. חזרות לא ייכללו בערכה. לאחר מכן, במיון בערכה, באמצעות שיטת הרשימה `count()`, נגלה את הכמות של כל ערך ברשימה המקורית. אם המספר הזה הוא 1, הרכיב הוא ייחודי.

תרגיל מס' 16.

מצא מספר שמופיע ברשימה בתדירות הגבוהה ביותר.

תרגיל מס' 17.

כתוב תוכנית שעושה חיפוש בינארי ברשימה מסודרת ותדפיס באיזה מקום נמצא איבר.

חיפוש בינארי של איבר.

חיפוש בינארי, המחפש ערך ברשימה או במערך משמש רק עבור רצפים מסודרים, כלומר ממוינים בסדר עולה או יורד. זה מורכב בקביעת האם המערך מכיל את הערך הרצוי, כמו גם בקביעת המיקום שלו.

תיאור האלגוריתם

מצא האלמנט האמצעי ברצף. לשם כך, המדדים הראשונים והאחרונים קשורים למשתנים, ומחושב האינדקס של האלמנט האמצעי.

הערך של האלמנט האמצעי מושווה לערך הרצוי. תלוי אם הוא גדול או קטן מערך האלמנט האמצעי, חיפוש נוסף יתבצע רק בצד שמאל או רק בחצאי הימני של המערך. אם הערך של האלמנט האמצעי שווה לערך הרצוי, החיפוש יסתיים.

אחרת, אחד הגבולות של הרצף הנלמד מוזז. אם הערך הרצוי גדול מערכו של האלמנט האמצעי, הגבול התחתון מועבר לפני אלמנט האמצע על ידי אלמנט אחד לימין. אם הערך הרצוי הוא פחות מערך האלמנט האמצעי, הגבול העליון מועבר על ידי האלמנט לפני האמצע.

שוב, האלמנט האמצעי נמצא כעת במחצית הנבחרת. האלגוריתם המתואר לעיל חוזר על עצמו עבור פרוסה זו.

תרגיל מס' 18.

תסדר רשימה בעזרת מיון בחירה (מיון מינימום או מקסימום).

אלגוריתם מיון הבחירה מורכב בחיפוש על פרוסה לא מעובדת. מחפשים במערך או רשימה ערך המינימלי ואז מחליפים ערך זה עם האלמנט הראשון של הפרוטה הלא מעובדת. בשלב הבא, הפרוטה הגולמית מצטמצמת באלמנט אחד.

האלגוריתם הוא:

מצא את הערך הקטן ביותר ברשימה.

כתוב אותו לראש הרשימה, ואת האלמנט הראשון למקום בו היה הקטן ביותר היה פעם.

שוב מצא את הפריט הקטן ביותר ברשימה. האלמנט הראשון אינו מעורב בחיפוש.

שים את המספר השני הקטן ביותר במקום השני ברשימה. האלמנט השני בו זמנית עובר למקום הפנוי.

המשך לחפש ולשתף עד שתגיע לסוף הרשימה.

תרגיל מס' 19.

תסדר רשימה בעזרת מיון בועות.

מיון בועות הוא שיטה למיון מערכים ורשימות על ידי השוואה והחלפה ברציפות של אלמנטים שכנים אם הקודם גדול מהנוכחי או להיפך. במהלך ביצוע אלגוריתם זה מופיעים בסוף הרשימה אלמנטים עם ערכים גדולים יותר, ואלמנטים עם ערכים נמוכים נעים בהדרגה לקראת תחילת הרשימה או הפוך. מבחינה פיגורטיבית, יסודות כבדים נופלים לתחתית, והריאות צפות לאט כמו בועות אוויר.

במיון הבועות, מספר האיטרציות של הלולאה החיצונית נקבע על ידי הרשימה הארוכה מינוס אחת, מכיוון שכאשר האלמנט השני נופל במקומו, הראשון הוא מינימלי או מקסימלי חד משמעי והוא נופל במקומו.

מספר החזרות של הלולאה הפנימית תלוי במספר האיטרציות של הלולאה החיצונית, שכן סוף הרשימה כבר ממוינת, וזה לא הגיוני לעבור על האלמנטים הללו.

לדוגמה הרשימה הבאה [6, 12, 4, 3, 8].

במהלך האיטרציה הראשונה של הלולאה החיצונית, המספר 12 עובר עד הסוף. זה ידרוש 4 השוואות בלופ הפנימי:

6 < 12? לא

12 < 4? כן. להחליף

12 < 3? כן. להחליף

12 < 8? כן. להחליף

תוצאה: [6, 4, 3, 8, 12]

במהלך האיטרציה השנייה של הלולאה החיצונית, מספר 8 עובר למקום אחד לפני אחרון. זה ידרוש 3 השוואות:

6 < 4? כן. להחליף

6 < 3? כן. להחליף

6 < 8? לא

תוצאה: [4, 3, 6, 8, 12]

באיטרציה השלישית של הלולאה החיצונית, שני האלמנטים האחרונים אינם נכללים. מספר החזרות של הלולאה הפנימית שווה לשניים:

4 < 3? כן. להחליף

4 < 6? לא

תוצאה: [3, 4, 6, 8, 12]

באיטרציה הרביעית של הלולאה החיצונית נותר להשוות רק את שני האלמנטים הראשונים, כך שמספר האיטרציות של הפנימי שווה לאחד:

3 < 4? לא

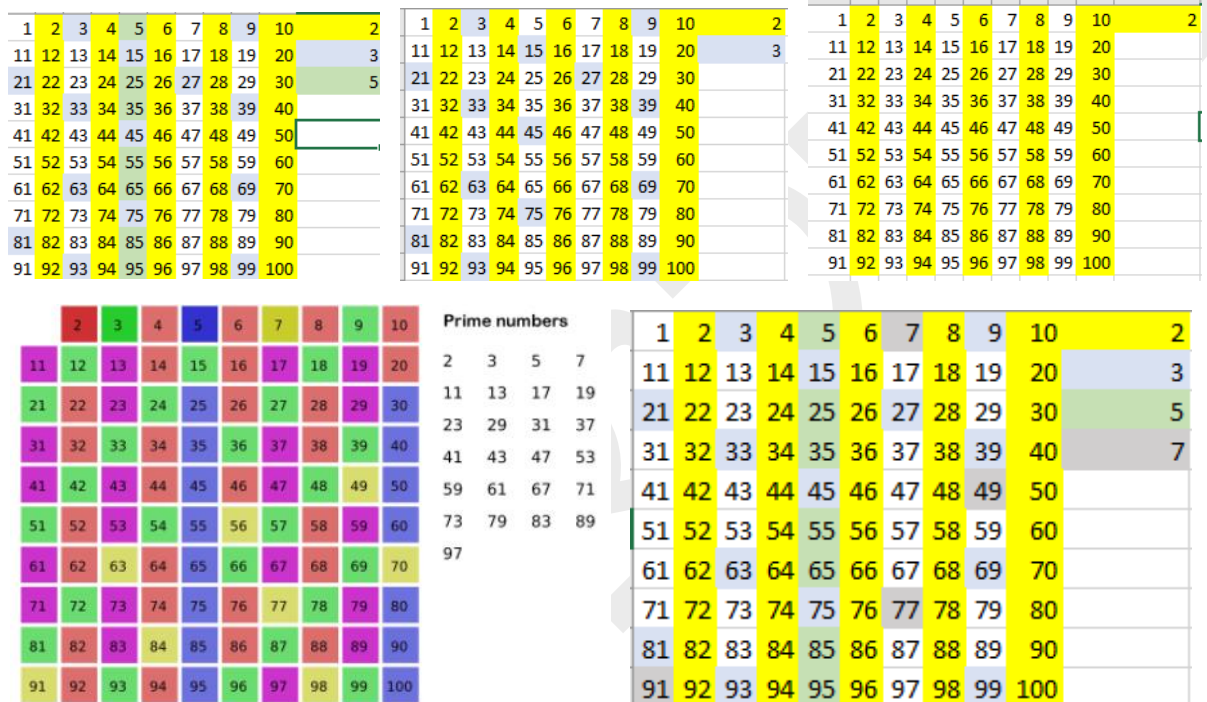
תוצאה: [3, 4, 6, 8, 12]

(ניתן לשדרג את התוכנה כך, שברגע שהרשימה מסודרת, אנו מפסיקים את המיון. לצורך זה נגדיר משתנה flag וניתן לו ערך 0 כל פעם שניכנס ללולאה הפנימית. במידה וביצענו החלפה, נשנה את

הדגל ל-1. בסיום הלולאה הפנימית נבדוק האם הדגל השתנה ל-1 או שנשאר 0. אם נשאר 0 זה אומר שלא הייתה אף החלפה ומערך מסודר ואז נצא מהלולאה החיצונית).

תרגיל מס' 20.

מצא כל המספרים הראשונים עד מספר N שנקלט מהמשתמש בעזרת שיטת "רשת".  
המסננת של ארטוסתנס - Sieve of Eratosthenes - שיטה למציאת מספרים ראשוניים.



המסננת של Eratosthenes היא אלגוריתם למציאת מספרים ראשוניים טבעיים על ידי ביטול בהדרגה של מספרים לא ראשוניים. באופן פיגורטיבי, מספרים מורכבים מחליקים דרך מסננת Eratosthenes במהלך טלטולם, בעוד שמספרים ראשוניים נשארים במסננת.

כדי להבין את האלגוריתם הזה, זכור כי המספרים הם ראשוניים אם הם מחולקים ללא שארית רק על 1 ועל עצמם. המספר הראשוני הראשון הוא 2, המספר הראשוני השני הוא 3.

כל המספרים השווים, למעט השניים, הם לא ראשוניים, כלומר הם אינם ראשוניים, מכיוון שהם מחולקים לא רק לעצמם וליחידה, אלא גם ל-2.

כל המספרים שהם כפול של שלושה, למעט המשולש עצמו, הם מורכבים, מכיוון שהם מחולקים לא רק לעצמם וליחידה, אלא גם ל-3.

המספר 4 כבר צנח מהמשחק, מכיוון שהוא מחולק ל-2.

המספר 5 הוא ראשוני, מכיוון שאף מחלוקת ראשית לא עומדת לפניו.

אם המספר אינו מתחלק למספר ראשוני שקדם לו, אז לא ניתן לחלק אותו במספר מורכב שקדם לו.

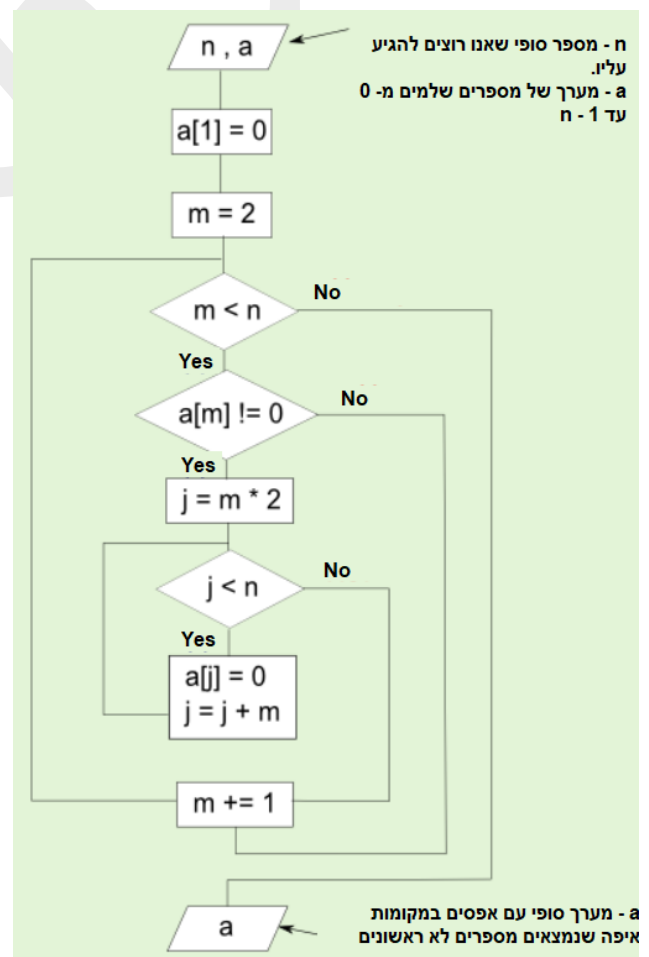
הנקודה האחרונה נובעת מהעובדה שתמיד ניתן לייצג מספרים מורכבים כתוצר של ראשוניות. לכן, אם מספר מורכב אחד מחולק על ידי קומפלקס אחר, אז הראשון צריך להיות מחולק על ידי מחלקים של השני. לדוגמה, 12 מחולק על ידי 6, המחלקים הם 2 ו-3. המספר 12 מחולק על ידי 2 ו-3.

אלגוריתם Eratosthenes מורכב במדויק בבדיקה ברצף של חלוקת המספרים לפי פריטים ראשוניים. ראשית, ראש הממשלה נלקח וכל הכפולים בו נזרעים ממספר מספרים טבעיים. ואז נלקח הפשוט הבא וכל הכפולים בו מחוסלים וכן הלאה.

כאשר מיישמים את אלגוריתם Eratosthenes בשפת תכנות, יש מורכבות מסוימת. נניח שאנחנו שמים מספרים טבעיים עד למספר נתון  $n$  במערך. בנוסף, במהלך ביצוע האלגוריתם, נחליף את המספרים המורכבים שהתגלו באפסים. לאחר ביצוע האלגוריתם, אותם תאים במערך שאינם מכילים אפסים מכילים מספרים ראשוניים המוצגים על המסך.

עם זאת, אינדקס מערך מתחיל באפס, ומספרים ראשוניים מתחילים בשניים. בעיה זו ניתנת לפתרון, אך מוסיפה מורכבות לקוד. מכיוון שאלגוריתם Eratosthenes אינו פשוט כל כך, קל יותר להזניח את ההתחלה ולקחת מערך מ-0 ל- $n$ . אינדקסים חשובים יותר כאן מערכי אלמנטים. ערכים יכולים להיות נכונים, מה שמצביע על מספר ראשוני, ו-False, שמציין מספר מורכב.

בדוגמה זו ליישום אלגוריתם Eratosthenes, הרשימה מלאה במספרים מ-0 עד  $n$  כולל כך שמדדי האלמנטים חופפים לערכיהם. יתר על כן, כל המספרים המורכבים מוחלפים על ידי אפסים:



תרגיל מס' 21.

כתוב תוכנית שמבקשת מהמשתמש להכניס 8 ערכים ומעתיקה אותם לתוך הרשימה. תוכנית מחפשת ערך מינימלי, ערך מקסימלי וסכום האיברים של הרשימה.

תרגיל מס' 22.

כתוב תוכנית שמייצרת 100 מספרים אקראיים ומעתיקה אותם לתוך הרשימה. התוכנית מדפיסה אותם בצורה הבאה: 10 שורות בכל שורה 10 איברים. אחרי זה תוכנית מסדרת את הרשימה מהערך הגדול ביותר לערך הקטן ביותר ושוב מדפיסה את הרשימה באותה צורה כמו בתחילת התרגיל. להדפסה יש לכתוב פונקציה שמקבלת את השם של רשימה ומדפיסה את הרשימה.

תרגיל מס' 23.

כתוב פונקציה שממלאת את הרשימה במספרים אקראיים. הפונקציה מקבלת את שם הרשימה, מספר איברים, מספר מינימלי של הטווח, מספר מקסימלי של הטווח. הפונקציה מדפיסה את המספרים שנמצאים ברשימה.

תרגיל מס' 24.

כתוב פונקציה שמבצעת חיפוש בינארי בתוך הרשימה. מקבלת את שם הרשימה וערך של חיפוש. הפונקציה מחזירה את המיקום של איבר הזה.

כתוב פונקציה שמבצעת הזזה בתוך הרשימה לפי כמות הצעדים שהשתמש מכניס. (הזזה צריכה להיות מעגלית).

תרגיל מס' 25.

בצע שינוי מחזורי ברשימת המספרים השלמים לפי מספר הצעדים שצוין. ההזזה צריכה להיות גם מעגלית, כלומר אלמנט החורג מהרשימה חייב להופיע מקצהו האחר.

כדי לפתור בעיה זו, יש להשתמש באחת השיטות הבאות מהרשימה:

append() - מוסיף פריט בסוף הרשימה,

insert() - מכניס אלמנט לאינדקס שצוין,

pop() - מאחזר אלמנט מסוף הרשימה או, אם עבר אינדקס, לפי אינדקס.

יש לתת לפונקציה shift() לקבל את הרשימה ומספר שלבי המשמרת כארגומנטים. אם הצעדים מיוצגים על ידי מספר חיובי, התזוזה מבוצעת ימינה, כלומר יש לחלץ את האלמנט האחרון ולהוסיף אותו להתחלה.

אם הצעדים הם מספר שלילי, אנו נבצע מעבר מימין לשמאל, כלומר עלינו לחלץ את האלמנט הראשון ולהוסיף אותו עד הסוף.

תרגיל מס' 26.

ממלאים רשימה במספרים אקראיים. צריכים לחשב כמה פעמים מופיע כל מספר. יש לממש פונקציה למילוי מערך ופונקציה למציאת מספרים.

כמות איברים זהים ברשימה.

נתונה רשימה של מספרים שלמים. חשב כמה פעמים כל מספר מופיע בו. לדוגמה, אם ניתנת רשימה [1, 1, 2, 3, 1, 4], בתוכה המספר 1 מופיע שלוש פעמים, המספר 3 - פעמיים, המספרים 2 ו-4 - פעם אחת.



שתהיה פונקציה בתוכנית הממלאת את הרשימה במספרים אקראיים בטווח ובכמות שצוינו על ידי המשתמש.

ניתן לממש את המשימה בעזרת רשימה שבכל מקום זוגי נמצא מספר, ובמקום אי-זוגי מספר שמציין כמה פעמים מופיע מספר, או בעזרת 2 רשימות, כאשר רשימה ראשונה מכילה מספרים ורשימה שנייה מכילה מספר שמציין כמה פעמים מופיע מספר.

תרגיל מס' 27.

כתוב תוכנית שמקבלת מהמשתמש רשימה (עד למספר 0) ומעתיק אותה לרשימה חדשה בסדר הפוך. יש להדפיס רשימה מקורית ורשימה סופית. (פתור ללא פונקציה מיוחדת של טיפול ברשימות ועם פונקציה מיוחדת).

תרגיל מס' 28.

תגדיר 2 רשימות בעל 10 מקומות. תכניס לכל רשימה ערכים אקראיים מ-0 עד 100, כך שכל מספר בתוך רשימה יהיה ייחודי. תאחד 2 רשימות האלה לרשימה שלישית כך שאם קיים אותו מספר גם ברשימה ראשונה וגם בשנייה, מעתיקים רק פעם אחד את המספר. לדוגמא:

$A=[1, 5, 2, 12, 34, 3, 37, 43, 10, 19]$

$B=[11, 25, 32, 12, 43, 3, 1, 8, 2, 9]$

$C=[1, 5, 2, 12, 34, 3, 37, 43, 10, 19, 11, 25, 32, 8, 2, 9]$

תרגיל מס' 29.

כתוב תוכנית שמקבלת מהמשתמש רשימה של 10 איברים והופכת כל אחד מ-5 זוגות בצורה הבאה:

$A=[1, 5, 2, 12, 34, 3, 37, 43, 10, 19] \rightarrow A=[5, 1, 12, 2, 3, 34, 43, 37, 19, 10]$

## פרק ז'. מערך Array.

### מה זה מערך ומה הבדל בין רשימה למערך.

מערכים ורשימות משמשים בפיתוח לאחסון נתונים, אך הם אינם משרתים את אוהצג המטרות בדיוק. ניתן להשתמש בשניהם לאחסון כל סוג נתונים (מספרים אמיתיים, מחרוזות וכו'), ושניהם ניתנים לאינדוקס, אך הדמיון בין השניים לא מרחיק לכת. ההבדל העיקרי בין רשימה למערך הוא הפונקציות שתוכלו לבצע בהן. לדוגמה, ניתן לחלק מערך ב-3, וכל מספר במערך יחולק ב-3 והתוצאה תודפס אם תבקש זאת. אם תנסה לחלק רשימה פי 3, מתורגמן הפיתוח יגיד לך שלא ניתן לעשות את זה ותתקבל שגיאה.

קיימת דרישה לצעד נוסף לשימוש במערכים מכיוון שצריך להכריז עליהם בעוד שרשימות לא, מכיוון שהן חלק מהתחביר של פיתוח, ולכן לרוב משתמשים ברשימות לעתים קרובות יותר, מה שעובד מצוין רוב הזמן. עם זאת, אם בכוונתך לבצע פונקציות חשבוניות על הרשימות שלך, יש להשתמש במערכים במקום זאת. בנוסף, מערכים יאחסנו את הנתונים בצורה קומפקטית ויעילה יותר, כך שאם מאחסנים כמות גדולה של נתונים, ניתן לשקול להשתמש גם במערכים.

דוגמה להגדרת המערך:

```
>>> x = array('i',[3, 6, 9, 12])
>>> print(x)
array('i', [3, 6, 9, 12])
>>> x = array([3, 6, 9, 12])
Traceback (most recent call last):
  File "<pyshell#7>", line 1, in <module>
    x = array([3, 6, 9, 12])
TypeError: array() argument 1 must be a unicode character, not list
```

מודול מערך בשפת פיתוח מגדיר סוג אובייקט שיכול לייצג באופן קומפקטי מערך של ערכים בסיסיים שלמים. מערכים הם סוגי רצפים ומתנהגים מאוד כמו רשימות, אלא שסוג האובייקטים המאוחסנים בהם מוגבל ויכול להיות רק מסוג אחד.

מה משותף בין רשימה לבין מערך:

- שניהם משמשים לשמירת נתונים.
- שניהם ניתנים לשינוי.
- שניהם ממוספרים וניתנים למעבר בין איבר לבין איבר.
- שניהם ניתנים לחיתוך ולחלוקה. (slice)

הבדל ביניהם:

ההבדל העיקרי בין שני סוגי הנתונים הללו הוא הפעולה שניתן לבצע עליהם. מערכים מותאמים במיוחד לחישובים אריתמטיים, כך שאם בכוונתנו לבצע פעולות דומות, יש לשקול להשתמש במערך במקום ברשימה.

כמו כן, רשימות הן מכולות עבור אלמנטים עם סוגים שונים של נתונים, אך מערכים משמשים כמכולות עבור אלמנטים מאותו סוג נתונים.

הדוגמה למטה היא תוצאה של חלוקת מערך במספר מסוים ועבור רשימה. כאשר אנו מנסים את אותה פעולה (דוגמה: חלוקה) ברשימה, אנו מקבלים TypeError מכיוון שרשימות פיתוח מובנות אינן טסליצקי מרק

תומכות בפרוטוקול `\_\_div\_\_`. צריך לבצע צעד נוסף כדי לבצע חישוב זה ברשימה מכיוון שאז תצטרך לעבור על כל פריט בזה אחר זה ולשמור ברשימה אחרת.

Array זה לא מבנה נתונים בנוי בפנים – כדי לאפשר עבודה עם סוג נתונים הזה צריכים לכתוב:

```
import array
```

בהגדרת המערך במקום ראשון צריכים לכתוב סוג של נתונים ואז איברים של מערך:

דוגמאות:

```
from array import *
array_num = array('i', [1,3,5,7,9,2])
for i in array_num:
    print(i,end=" ")
print("Access first three items individually")
print(array_num[0])
print(array_num[1])
print(array_num[2])
```

```
1 3 5 7 9 2 Access first three items individually
1
3
5
```

אותה דוגמה עם איברים מסוגים שונים:

```
from array import *
array_num = array('i', [1,3,5,7,9,"2"])
for i in array_num:
    print(i,end=" ")
print("Access first three items individually")
print(array_num[0])
print(array_num[1])
print(array_num[2])
```

```
Traceback (most recent call last):
  File "C:/MyPython/targil.py", line 2, in <module>
    array_num = array('i', [1,3,5,7,9,"2"])
TypeError: an integer is required (got type str)
```

או אפילו אם נוסיף איבר עם floating point, נקבל גם שגיאה:

```
from array import *
array_num = array('i', [1,3,5,7,9,2.0])
for i in array_num:
    print(i,end=" ")
print("Access first three items individually")
print(array_num[0])
print(array_num[1])
print(array_num[2])
```

```
Traceback (most recent call last):
  File "C:/MyPython/targil.py", line 2, in <module>
    array_num = array('i', [1,3,5,7,9,2.0])
TypeError: integer argument expected, got float
```

```
from array import *
array_num = array('f', [1.2,3,5.8,7,9,2.0])
for i in array_num:
    print(i,end=" ")
print("Access first three items individually")
print(array_num[0])
print(array_num[1])
print(array_num[2])
```

```
1.2000000476837158 3.0 5.800000190734863 7.0 9.0 2.0 Access first three items individually
1.2000000476837158
3.0
5.800000190734863
```

```
from array import *
array_num = array('u', ["1","2","3","s","5","8","7","9","2"])
for i in array_num:
    print(i,end=" ")
print("Access first three items individually")
print(array_num[0])
print(array_num[1])
print(array_num[2])
```

```
1 2 3 s 5 8 7 9 2 Access first three items individually
1
2
3
```

טבלה של סוגי איברים של מערך:

גודל בבתים	משתנה דומה בשפת C	סוג המשתנה	אות להגדרת סוג המערך
1	signed char	int	'b'
1	unsigned char	int	'B'
2	Py_UNICODE	Unicode character	'u'
2	signed short	int	'h'
2	unsigned short	int	'H'
2	signed int	int	'i'
2	unsigned int	int	'I'
4	signed long	int	'l'
4	unsigned long	int	'L'
8	signed long	int	'q'
8	unsigned long	int	'Q'
4	float	float	'f'
8	double	float	'd'

## פונקציות מיוחדות של מערכים:

פונקציה `append()` מאפשרת להוסיף למערך איבר של אותו סוג בסוף המערך.

נניח ואנו רוצים להוסיף איבר חדש למערך קיים:

```
from array import *
array_num = array('i', [1, 3, 5, 7, 9])
print("Original array: "+str(array_num))
print("Append 11 at the end of the array:")
array_num.append(11)
print("New array: "+str(array_num))
```

```
Original array: array('i', [1, 3, 5, 7, 9])
Append 11 at the end of the array:
New array: array('i', [1, 3, 5, 7, 9, 11])
```

נניח ורוצים להפוך את הסדר של האיברים במערך:

```
from array import *
array_num = array('i', [1, 3, 5, 3, 7, 1, 9, 3])
print("Original array: "+str(array_num))
array_num.reverse()
print("Reverse the order of the items:")
print(str(array_num))
```

```
Original array: array('i', [1, 3, 5, 3, 7, 1, 9, 3])
Reverse the order of the items:
array('i', [3, 9, 1, 7, 3, 5, 3, 1])
```

נראה כמה בתים לוקח כל איבר של מערך:

```
from array import *
array_num = array('i', [1, 3, 5, 7, 9])
print("Original array: "+str(array_num))
print("Length in bytes of one array item: "+str(array_num.itemsize))
```

```
Original array: array('i', [1, 3, 5, 7, 9])
Length in bytes of one array item: 4
```

נראה דוגמה לקבלת מיקום בזיכרון ואורך של חוצץ שמחזיק את המערך.

```
from array import *
array_num = array('i', [1, 3, 5, 7, 9])
print("Original array: "+str(array_num))
print("Current memory address and the length in elements of the buffer: "
+str(array_num.buffer_info()))
print("The size of the memory buffer in bytes: " +str(array_num.buffer_info()[1] *
array_num.itemsize))
```

```
Original array: array('i', [1, 3, 5, 7, 9])
Current memory address and the length in elements of the buffer: (46393032, 5)
The size of the memory buffer in bytes: 20
```

נראה תוכנית שבודקת כמה פעמים מופיע איבר בתוך המערך.

```
from array import *
array_num = array('i', [1, 3, 5, 3, 7, 9, 3])
print("Original array: "+str(array_num))
print("Number of occurrences of the number 3 in the said array: "+str(array_num.count(3)))
```

```
Original array: array('i', [1, 3, 5, 3, 7, 9, 3])
Number of occurrences of the number 3 in the said array: 3
```

נראה דוגמה לחיבור איברים של מערך בסוף המערך. (שכפול המערך).

```
from array import *
array_num = array('i', [1, 3, 5, 7, 9])
```

```
print("Original array: "+str(array_num))
array_num.extend(array_num)
print("Extended array: "+str(array_num))
```

```
Original array: array('i', [1, 3, 5, 7, 9])
Extended array: array('i', [1, 3, 5, 7, 9, 1, 3, 5, 7, 9])
```

כיצד אפשר להפוך מספרים שלמים של המערך למחרוזת תווים.

```
from array import *
print("Bytes to String: ")
x = array('b', [115, 111, 117, 114, 99, 101])
s = x.tobytes()
print(s)
```

```
Bytes to String:
b'source'
```

נעתיק איברי הרשימה למערך.

```
from array import *
num_list = [1, 2, 6, -8]
array_num = array('i', [])
print("Items in the list: " + str(num_list))
print("Append items from the list: ")
array_num.fromlist(num_list)
print("Items in the array: "+str(array_num))
```

```
Items in the list: [1, 2, 6, -8]
Append items from the list:
Items in the array: array('i', [1, 2, 6, -8])
```

נכניס איבר חדש לפני איבר השני בתוך המערך.

```
from array import *
array_num = array('i', [1, 3, 5, 7, 9])
print("Original array: "+str(array_num))
print("Insert new value 4 before 3:")
array_num.insert(1, 4)
print("New array: "+str(array_num))
```

```
Original array: array('i', [1, 3, 5, 7, 9])
Insert new value 4 before 3:
New array: array('i', [1, 4, 3, 5, 7, 9])
```

נוריד את האיבר המסוים מהמערך בעזרת מיקום בתוך המערך.

```
from array import *
array_num = array('i', [1, 3, 5, 7, 9])
print("Original array: "+str(array_num))
print("Remove the third item form the array:")
array_num.pop(2)
print("New array: "+str(array_num))
```

```
Original array: array('i', [1, 3, 5, 7, 9])
Remove the third item form the array:
New array: array('i', [1, 3, 7, 9])
```

נמחק מתוך המערך את האיבר בפעם הראשונה שנראה אותו במערך.

```
from array import *
array_num = array('i', [1, 3, 5, 3, 7, 1, 9, 3])
print("Original array: "+str(array_num))
print("Remove the first occurrence of 3 from the said array:")
array_num.remove(3)
print("New array: "+str(array_num))
array_num.remove(3)
print("New array: "+str(array_num))
array_num.remove(3)
print("New array: "+str(array_num))
array_num.remove(3)
print("New array: "+str(array_num))
```

```
Original array: array('i', [1, 3, 5, 3, 7, 1, 9, 3])
Remove the first occurrence of 3 from the said array:
New array: array('i', [1, 5, 3, 7, 1, 9, 3])
New array: array('i', [1, 5, 7, 1, 9, 3])
New array: array('i', [1, 5, 7, 1, 9])
Traceback (most recent call last):
  File "C:/MyPython/targil.py", line 11, in <module>
    array_num.remove(3)
ValueError: array.remove(x): x not in array
```



ניתן לתקן דוגמה הקודמת בעזרת try except

```
from array import *
array_num = array('i', [1, 3, 5, 3, 7, 1, 9, 3])
print("Original array: "+str(array_num))
print("Remove the first occurrence of 3 from the said array:")
array_num.remove(3)
print("New array: "+str(array_num))
array_num.remove(3)
print("New array: "+str(array_num))
array_num.remove(3)
print("New array: "+str(array_num))
try:
    array_num.remove(3)
    print("New array: "+str(array_num))
except:
    print("No element")
```

```
Original array: array('i', [1, 3, 5, 3, 7, 1, 9, 3])
Remove the first occurrence of 3 from the said array:
New array: array('i', [1, 5, 3, 7, 1, 9, 3])
New array: array('i', [1, 5, 7, 1, 9, 3])
New array: array('i', [1, 5, 7, 1, 9])
No element
```

נראה את הדוגמה של הפיכת מערך לרשימה:

```
from array import *
array_num = array('i', [1, 3, 5, 3, 7, 1, 9, 3])
print("Original array: "+str(array_num))
num_list = array_num.tolist()
print("Convert the said array to an ordinary list with the same items:")
print(num_list)
```

```
Original array: array('i', [1, 3, 5, 3, 7, 1, 9, 3])
Convert the said array to an ordinary list with the same items:
[1, 3, 5, 3, 7, 1, 9, 3]
```

## תרגול בנושא מערכים. תרגול בנושא טבלאות מעקב.

תרגיל מס' 1.

תפענח את הקוד הבא. מה עושה תוכנית? מה יהיה פלט של הקוד?

```
from array import *
array_num = array('i', [1,3,5,7,9])
for i in array_num:
    print(i)
print("Access first three items individually")
print(array_num[0])
print(array_num[1])
print(array_num[2])
```

תרגיל מס' 2.

תפענח את הקוד הבא. מה עושה תוכנית? מה יהיה פלט של הקוד?

```
from array import *
array_num = array('i', [1, 3, 5, 7, 9])
print("Original array: "+str(array_num))
print("Append 11 at the end of the array:")
array_num.append(11)
print("New array: "+str(array_num))
```

תרגיל מס' 3.

תפענח את הקוד הבא. מה עושה תוכנית? מה יהיה פלט של הקוד?

```
from array import *
array_num = array('i', [1, 3, 5, 3, 7, 1, 9, 3])
print("Original array: "+str(array_num))
array_num.reverse()
print("Reverse the order of the items:")
print(str(array_num))
```

תרגיל מס' 4.

תפענח את הקוד הבא. מה עושה תוכנית? מה יהיה פלט של הקוד?

```
from array import *
array_num = array('i', [1, 3, 5, 7, 9])
print("Original array: "+str(array_num))
print("Length in bytes of one array item: "+str(array_num.itemsize))
```

תרגיל מס' 5.

תפענח את הקוד הבא. מה עושה תוכנית? מה יהיה פלט של הקוד?

```
from array import *
array_num = array('i', [1, 3, 5, 7, 9])
print("Original array: "+str(array_num))
print("Current memory address and the length in elements of the buffer:
"+str(array_num.buffer_info()))
print("The size of the memory buffer in bytes: "+str(array_num.buffer_info()[1] *
array_num.itemsize))
```

תרגיל מס' 6.

תפענח את הקוד הבא. מה עושה תוכנית? מה יהיה פלט של הקוד?

```
from array import *
array_num = array('i', [1, 3, 5, 3, 7, 9, 3])
print("Original array: "+str(array_num))
print("Number of occurrences of the number 3 in the said array: "+str(array_num.count(3)))
```

תרגיל מס' 7.

תפענח את הקוד הבא. מה עושה תוכנית? מה יהיה פלט של הקוד?

```
from array import *
array_num = array('i', [1, 3, 5, 7, 9])
print("Original array: "+str(array_num))
array_num.extend(array_num)
print("Extended array: "+str(array_num))
```

תרגיל מס' 8.

תפענח את הקוד הבא. מה עושה תוכנית? מה יהיה פלט של הקוד?

```
from array import *
print("Bytes to String: ")
x = array('b', [119, 51, 114, 101, 115, 111, 117, 114, 99, 101])
s = x.tobytes()
print(s)
```

תרגיל מס' 9.

תפענח את הקוד הבא. מה עושה תוכנית? מה יהיה פלט של הקוד?

```
from array import *
num_list = [1, 2, 6, -8]
array_num = array('i', [])
print("Items in the list: " + str(num_list))
print("Append items from the list: ")
array_num.fromlist(num_list)
print("Items in the array: "+str(array_num))
```

תרגיל מס' 10.

תפענח את הקוד הבא. מה עושה תוכנית? מה יהיה פלט של הקוד?

```
from array import *
array_num = array('i', [1, 3, 5, 7, 9])
print("Original array: "+str(array_num))
print("Insert new value 4 before 3:")
array_num.insert(1, 4)
print("New array: "+str(array_num))
```

תרגיל מס' 11.

תפענח את הקוד הבא. מה עושה תוכנית? מה יהיה פלט של הקוד?

```
from array import *
array_num = array('i', [1, 3, 5, 7, 9])
print("Original array: "+str(array_num))
print("Remove the third item form the array:")
array_num.pop(2)
print("New array: "+str(array_num))
```

תרגיל מס' 12.

תפענח את הקוד הבא. מה עושה תוכנית? מה יהיה פלט של הקוד?

```
from array import *
array_num = array('i', [1, 3, 5, 3, 7, 1, 9, 3])
print("Original array: "+str(array_num))
print("Remove the first occurrence of 3 from the said array:")
array_num.remove(3)
print("New array: "+str(array_num))
```

תרגיל מס' 13.

תפענח את הקוד הבא. מה עושה תוכנית? מה יהיה פלט של הקוד?

```
from array import *
array_num = array('i', [1, 3, 5, 3, 7, 1, 9, 3])
print("Original array: "+str(array_num))
num_list = array_num.tolist()
print("Convert the said array to an ordinary list with the same items:")
print(num_list)
```

תרגיל מס' 14.

תפענח את הקוד הבא. מה עושה תוכנית? מה יהיה פלט של הקוד?

```
def test_duplicate(array_nums):
    nums_set = set(array_nums)
    return len(array_nums) != len(nums_set)
print(test_duplicate([1,2,3,4,5]))
print(test_duplicate([1,2,3,4, 4]))
print(test_duplicate([1,1,2,2,3,3,4,4,5]))
```

תרגיל מס' 15.

תפענח את הקוד הבא. מה עושה תוכנית? מה יהיה פלט של הקוד?

```
def find_first_duplicate(nums):
    num_set = set()
    no_duplicate = -1
    for i in range(len(nums)):
        if nums[i] in num_set:
            return nums[i]
        else:
            num_set.add(nums[i])
    return no_duplicate
print(find_first_duplicate([1, 2, 3, 4, 4, 5]))
print(find_first_duplicate([1, 2, 3, 4]))
print(find_first_duplicate([1, 1, 2, 3, 3, 2, 2]))
```

תרגיל מס' 16.

תפענח את הקוד הבא. מה עושה תוכנית? מה יהיה פלט של הקוד?

```
def max_Product(arr):
    arr_len = len(arr)
```

```

if (arr_len < 2):
    print("No pairs exists")
    return
x = arr[0]; y = arr[1]
for i in range(0, arr_len):
    for j in range(i + 1, arr_len):
        if (arr[i] * arr[j] > x * y):
            x = arr[i]; y = arr[j]
    return x,y
nums = [1, 2, 3, 4, 7, 0, 8, 4]
print("Original array:", nums)
print("Maximum product pair is:", max_Product(nums))
nums = [0, -1, -2, -4, 5, 0, -6]
print("\nOriginal array:", nums)
print("Maximum product pair is:", max_Product(nums))

```

תרגיל מס' 17.

תפענח את הקוד הבא. מה עושה תוכנית? מה יהיה פלט של הקוד?

```

from array import array
my_array = array('i', [10, 20, 30, 40, 50])
for i in my_array:
    print(i)

```

תרגיל מס' 18.

תפענח את הקוד הבא. מה עושה תוכנית? מה יהיה פלט של הקוד?

```

from array import array
a = array("I", (12,25))
print("Array buffer start address in memory and number of elements.")
print(a.buffer_info())

```

תרגיל מס' 19.

תפענח את הקוד הבא. מה עושה תוכנית? מה יהיה פלט של הקוד?

```

from array import array
num_array = array('i', [10,20,30,40,50])
print("Length of the array is:")
print(len(num_array))

```

תרגיל מס' 20.

תפענח את הקוד הבא. מה עושה תוכנית? מה יהיה פלט של הקוד?

```
from array import array
a = array("I", (12,25))
print(a.itemsize)
a = array("f", (12.236,36.36))
print(a.itemsize)
```

תרגיל מס' 21.

תפענח את הקוד הבא. מה עושה תוכנית? מה יהיה פלט של הקוד?

```
from array import array
import binascii
array1 = array('i', [7, 8, 9, 10])
print('array1:', array1)
as_bytes = array1.tobytes()
print('Bytes:', binascii.hexlify(as_bytes))
array2 = array('i')
array2.frombytes(as_bytes)
print('array2:', array2)
```

תרגיל מס' 22.

תפענח את הקוד הבא. מה עושה תוכנית? מה יהיה פלט של הקוד?

```
import array as arr
def test(nums):
    return sorted(set(nums),key=nums.index)
array_num = arr.array('i', [1, 3, 5, 1, 3, 7, 9])
print("Original array:")
for i in range(len(array_num)):
    print(array_num[i], end=' ')
print("\nAfter removing duplicate elements from the said array:")
result = arr.array('i', test(array_num))
for i in range(len(result)):
    print(result[i], end=' ')
array_num = arr.array('i', [2, 4, 2, 6, 4, 8])
print("\nOriginal array:")
for i in range(len(array_num)):
    print(array_num[i], end=' ')
print("\nAfter removing duplicate elements from the said array:")
result = arr.array('i', test(array_num))
```

```
for i in range(len(result)):
    print(result[i], end=' ')
```

תרגיל מס' 23.

תפענח את הקוד הבא. מה עושה תוכנית? מה יהיה פלט של הקוד?

```
import array as arr
def test(nums):
    return sum(range(10, 21)) - sum(list(nums))
array_num = arr.array('i', [10, 11, 12, 13, 14, 16, 17, 18, 19, 20])
print("Original array:")
for i in range(len(array_num)):
    print(array_num[i], end=' ')
print("\nMissing number in the said array (10-20): ",test(array_num))
array_num = arr.array('i', [10, 11, 12, 13, 14, 15, 16, 17, 18, 19])
print("\nOriginal array:")
for i in range(len(array_num)):
    print(array_num[i], end=' ')
print("\nMissing number in the said array (10-20): ",test(array_num))
```

תרגיל מס' 24.

תפענח את הקוד הבא. מה עושה תוכנית? מה יהיה פלט של הקוד?

```
import array as arr
a = arr.array('i', [1, 2, 3])
print ("Array before insertion : ", end = " ")
for i in range (0, 3):
    print (a[i], end = " ")
print()
a.insert(1, 4)
print ("Array after insertion : ", end = " ")
for i in (a):
    print (i, end = " ")
print()
b = arr.array('d', [2.5, 3.2, 3.3])
print ("Array before insertion : ", end = " ")
for i in range (0, 3):
    print (b[i], end = " ")
print()
b.append(4.4)
print ("Array after insertion : ", end = " ")
for i in (b):
    print (i, end = " ")
print()
```



תרגיל מס' 25.

תפענח את הקוד הבא. מה עושה תוכנית? מה יהיה פלט של הקוד?

```
import array
arr = array.array('i', [1, 2, 3, 1, 5])
print ("The new created array is : ", end = "")
for i in range (0, 5):
    print (arr[i], end = " ")
print ("\r")
print ("The popped element is : ", end = "")
print (arr.pop(2))
print ("The array after popping is : ", end = "")
for i in range (0, 4):
    print (arr[i], end = " ")
print("\r")
arr.remove(1)
print ("The array after removing is : ", end = "")
for i in range (0, 3):
    print (arr[i], end = " ")
```

תרגיל מס' 26.

תפענח את הקוד הבא. מה עושה תוכנית? מה יהיה פלט של הקוד?

```
import array as arr
l = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
a = arr.array('i', l)
print("Initial Array: ")
for i in (a):
    print(i, end = " ")
Sliced_array = a[3:8]
print("\nSlicing elements in a range 3-8: ")
print(Sliced_array)
Sliced_array = a[5:]
print("\nElements sliced from 5th "
      "element till the end: ")
print(Sliced_array)
Sliced_array = a[:]
print("\nPrinting all elements using slice operation: ")
print(Sliced_array)
```

תרגיל מס' 27.

תפענח את הקוד הבא. מה עושה תוכנית? מה יהיה פלט של הקוד?

```
import array
arr = array.array('i', [1, 2, 3, 1, 2, 5])
print ("The new created array is : ", end = "")
for i in range (0, 6):
    print (arr[i], end = " ")
print ("\r")
print ("The index of 1st occurrence of 2 is : ", end = "")
print (arr.index(2))
print ("The index of 1st occurrence of 1 is : ", end = "")
print (arr.index(1))
```

תרגיל מס' 28.

תפענח את הקוד הבא. מה עושה תוכנית? מה יהיה פלט של הקוד?

```
import array
arr = array.array('i', [1, 2, 3, 1, 2, 5])
print ("Array before updation : ", end = "")
for i in range (0, 6):
    print (arr[i], end = " ")
print ("\r")
arr[2] = 6
print("Array after updation : ", end = "")
for i in range (0, 6):
    print (arr[i], end = " ")
print()
arr[4] = 8
print("Array after updation : ", end = "")
for i in range (0, 6):
    print (arr[i], end = " ")
```

## תרגול בנושא כתיבת קוד.

תרגיל מס' 1.

כתוב תוכנית שמקבלת מהמשתמש מספר  $N$  שלם וחיובי ומגדירה מערך של מספרים אי-זוגיים לפי סדר עולה החל ממספר 1 בכמות האיברים ששווה ל- $N$ . בסוף בתוכנית יש להדפיס את המערך.

תרגיל מס' 2.

כתוב תוכנית שמקבלת מהמשתמש מספר  $N$  שלם וחיובי ומגדירה מערך של חזקות של 2 לפי סדר עולה החל ממספר 1 (2 בחזקת 0) בכמות האיברים ששווה ל- $N$ . בסוף בתוכנית יש להדפיס את המערך.

תרגיל מס' 3.

כתוב תוכנית שמקבלת איבר ראשון של סדרה חשבונית  $A$ , הפרש הסדרה  $D$  וכמות איברים של הסדרה  $N$ . תוכנה מגדירה מערך שמכיל איברים של הסדרה ומציגה אותו על המסך.

תרגיל מס' 4.

כתוב תוכנית שמקבלת איבר ראשון של סדרה הנדסית  $A$ , מנה של הסדרה  $Q$  וכמות איברים של הסדרה  $N$ . תוכנה מגדירה מערך שמכיל איברים של הסדרה ומציגה אותו על המסך.

תרגיל מס' 5.

כתוב תוכנית שמגדירה שני מספרים ראשונים של סדרת פיבונצ'י (0 ו-1), וכמות איברים של הסדרה  $N$  (גדול מ-2). תוכנה מגדירה מערך שמכיל איברים של הסדרה ומציגה אותו על המסך.

תרגיל מס' 6.

כתוב תוכנית שמקבלת שני מספרים ראשונים של הסדרה, וכמות איברים של הסדרה  $N$  (גדול מ-2). תוכנית מגדירה סדרה כך שאיבר הבא שווה לסכום של כל האיברים מהתחלה עד האיבר הנוכחי. תוכנית מגדירה מערך שמכיל איברים של הסדרה ומציגה אותו על המסך.

תרגיל מס' 7.

כתוב תוכנית שמקבלת מהמשתמש מערך בגודל 10 ומעתיקה איברים שלו למערך אחר בסדר הפוך. תוכנית מדפיסה את שני מערכים בסדר הפוך.

תרגיל מס' 8.

כתוב תוכנית שמקבלת מהמשתמש אורך המערך. מערך עצמו מקבלים מגנרטור של מספרים אקראיים (ערכים שלמים בלבד). תוכנית מדפיסה כל המספרים האי-זוגיים ומיקומם במערך בסדר עולה.

תרגיל מס' 9.

כתוב תוכנית שמקבלת מהמשתמש אורך המערך. מערך עצמו מקבלים מגנרטור של מספרים אקראיים (ערכים שלמים בלבד). תוכנית מדפיסה כל המספרים שמתחלקים ב-3 או ב-5 ומיקומם במערך בסדר יורד.

תרגיל מס' 10.

כתוב תוכנית שמקבלת מהמשתמש אורך המערך. מערך עצמו מקבלים מגנרטור של מספרים אקראיים (ערכים שלמים בלבד). תוכנית מדפיסה כל המספרים האי-זוגיים ומיקומם במערך בסדר עולה וכל הערכים הזוגיים ומיקומם בסדר יורד.

תרגיל מס' 11.

כתוב תוכנית שמקבלת מהמשתמש אורך המערך. מערך עצמו מקבלים מגנרטור של מספרים אקראיים (ערכים שלמים בלבד). תוכנית מדפיסה כל המספרים שעומדים במקומות האי-זוגיים במערך בסדר עולה וכל האיברים שעומדים במקומות הזוגיים בסדר יורד.

תרגיל מס' 12.

כתוב תוכנית שמקבלת מהמשתמש אורך המערך  $N$ . מערך עצמו מקבלים מגנרטור של מספרים אקראיים (ערכים שלמים בלבד). תוכנית מדפיסה איברים בצורה הבאה:  $A_1, A_N, A_2, A_{N-1}, A_3, A_{N-2}, \dots$

תרגיל מס' 13.

כתוב תוכנית שמקבלת מהמשתמש אורך המערך  $N$ . מערך עצמו מקבלים מגנרטור של מספרים אקראיים (ערכים שלמים בלבד). תוכנית מדפיסה איברים בצורה הבאה:

$A_1, A_2, A_N, A_{N-1}, A_3, A_4, A_{N-2}, A_{N-3}, \dots$

תרגיל מס' 14.

כתוב תוכנית שמקבלת מערך בעל 10 מספרים שלמים (חיוביים ושליילים) ומספר חיובי  $A$  מהמשתמש. תוכנית צריכה להציג את כל האיברים שבערך מוחלט יותר גדולים ממספר  $A$ .

תרגיל מס' 15.

כתוב תוכנית שמקבלת מערך בעל 10 מספרים שלמים (חיוביים ושלייליים) ומספרים חיוביים A ו-B מהמשתמש. תוכנית צריכה להציג את כל האיברים שבערך מוחלט יותר גדולים ממספר A ויותר קטן ממספר B.

תרגיל מס' 16.

כתוב תוכנית שמקבלת מערך בעל 10 מספרים שלמים (חיוביים ושלייליים) ומספרים חיוביים A ו-B מהמשתמש. תוכנית צריכה להציג את כל האיברים שבערך מוחלט יותר גדולים ממספר A ויותר קטן ממספר B ולחשב את הסכום שלהם.

תרגיל מס' 17.

כתוב תוכנית שמקבלת מערך בעל 10 מספרים שלמים (חיוביים ושלייליים) ובודקת האם מערך מסודר בסדר עולה?

תרגיל מס' 18.

כתוב תוכנית שמקבלת מערך בעל 10 מספרים שלמים (חיוביים ושלייליים) ובודקת האם מערך מסודר בסדר יורד?

תרגיל מס' 19.

כתוב תוכנית שמקבלת מערך בעל 10 מספרים שלמים (חיוביים ושלייליים) ובודקת האם מערך מסודר לפי סדרה חשבונית? אם כן, מדפיסים הפרש של סדרה, אם לא מדפיסים הודעה מתאימה.

תרגיל מס' 20.

כתוב תוכנית שמקבלת מערך בעל 10 מספרים שלמים (חיוביים ושלייליים) ובודקת האם מערך מסודר לפי סדרה הנדסית? אם כן, מדפיסים מנה של סדרה, אם לא מדפיסים הודעה מתאימה.

תרגיל מס' 21.

כתוב תוכנית שמקבלת מערך בעל 10 מספרים שלמים (חיוביים ושלייליים) ובודקת האם במערך בכל מקום זוגי עומד מספר זוגי ובכל מקום אי-זוגי עומד מספר אי-זוגי. במידה וכן, תדפיס הודעה מתאימה, אם לא מדפיסים את המספרים שלא נמצאים במקומות שלהם.

תרגיל מס' 22.

כתוב תוכנית שמקבלת מערך בעל 10 מספרים שלמים (חיוביים ושלייליים) ומוציאה מספר מינימלי שעומד במקום זוגי, ומספר מקסימלי שעומד במקום האי-זוגי.

תרגיל מס' 23.

כתוב תוכנית שמקבלת מערך בעל 10 מספרים שלמים (חיוביים ושלייליים) ומחפשת איברים שיותר גדולים מהשכן מצד ימין. יש להדפיס את כל הזוגות שמתאימים לקריטריון הזה.

תרגיל מס' 24.

כתוב תוכנית שמקבלת מערך בעל 10 מספרים שלמים (חיוביים ושלייליים) ומחפשת איברים שיותר גדולים מהשכן מצד ימין ויותר קטנים מהשכן מצד שמאל. יש להדפיס את כל השלישיות שמתאימות לקריטריון הזה.

תרגיל מס' 25.

כתוב תוכנית שמקבלת מערך בעל 10 מספרים שלמים (חיוביים ושלייליים) ומחפשת איברים שיותר גדולים מהמוצע של כל האיברים של מערך.

תרגיל מס' 26.

כתוב תוכנית שמקבלת מערך בעל 20 מספרים שלמים חיוביים אקראיים ומחפשת כמה רצפים של מספרים עולים יש במערך. יש להדפיס את כל הרצפים, לדוגמה:

51 31 63 52 3 10 5 56 81 73 9 32 71 25 89 61 71 83 97 16

31 63

3 10

5 56 81

9 32 71

25 89

61 71 83 97

תרגיל מס' 27.

כתוב תוכנית שמקבלת מערך בעל 10 מספרים שלמים חיוביים אקראיים ומחפשת את המרחק הקטן ביותר (בערך מוחלט) בין שכנים ומדפיסה את הזוג הזה. לדוגמה:

51 31 63 52 3 10 5 56 81 73

10 5 difference is 5

תרגיל מס' 28.

כתוב תוכנית שמקבלת מערך בעל 10 מספרים שלמים חיוביים אקראיים ומחפשת את הסכום הקטן ביותר וסכום הקטן ביותר בין שכנים ומדפיסה את הזוגות האלה. לדוגמה:

51 31 63 52 3 10 5 56 81 73

Max sum is 81 73, Min sum is 3 10

תרגיל מס' 29.

כתוב תוכנית שמקבלת מערך בעל 10 מספרים שלמים חיוביים אקראיים ובונה מערך חדש בעל 10 מקומות, שסכום של זוג מספרים ממערך מקורי נמצא במקומות זוגיים (0,2,4,6,8) והפרש במקומות אי-זוגיים (1,3,5,7).

תרגיל מס' 30.

כתוב תוכנית שמקבלת מערך בעל 20 מספרים שלמים חיוביים אקראיים ומחפשת את כל המספרים שחוזרים על עצמם. תוכנית צריכה להדפיס את המספר ואת מיקומם במערך. לדוגמה:

1 5 3 61 4 7 1 3 4 3 2 5 1 19 4 3 5 16 11 2

1 - 0, 7, 13 ...

תרגיל מס' 31.

כתוב תוכנית שמקבלת מערך בעל 20 מספרים שלמים חיוביים אקראיים ומחפשת את כל המספרים שנמצאים רק פעם אחד במערך. לדוגמה:

1 5 3 61 4 7 1 3 4 3 2 5 1 19 4 3 5 16 11 2

61 7 19 16 11

תרגיל מס' 32.

כתוב תוכנית שמקבלת מערך בעל 20 מספרים שלמים חיוביים אקראיים. תוכנית צריכה להדפיס את כל המספרים רק פעם אחד. לדוגמה:

1 5 3 61 4 7 1 3 4 3 2 5 1 19 4 3 5 16 11 2

1 5 3 61 4 7 2 19 16 11

תרגיל מס' 33.

כתוב תוכנית שמקבלת 2 מערכים בעל 10 מספרים כל אחד (שלמים חיוביים אקראיים) ומעתיקה אותם למערך השלישי כך שכל מקום זוגי תופס איבר ממערך A ומקום אי-זוגי איבר ממערך B. לדוגמה:

1 5 3 61 4 7 1 3 4 3

2 5 1 19 4 3 5 16 11 2

1 2 5 5 3 1 61 19 4 4 7 3 1 5 3 16 4 11 3 2

תרגיל מס' 34.

כתוב תוכנית שמקבלת 2 מערכים בעל 10 מספרים כל אחד (שלמים חיוביים אקראיים), מסדרת אותם בסדר עולה ומעתיקה אותם למערך השלישי כך שגם מערך השלישי יהיה מסודר בסדר עולה. (שימו לב שסידור של מערך השלישי יש לסדר תוך כדי העתקה).

תרגיל מס' 35.

כתוב תוכנית שמקבלת מערך בעל 20 מספרים שלמים חיוביים אקראיים. תוכנית צריכה להחליף את כל האיברים שנמצאים במקומות זוגיים אם כל האיברים שנמצאים במקומות אי-זוגיים.

תרגיל מס' 36.

כתוב תוכנית שמקבלת מערך בעל 20 מספרים שלמים חיוביים אקראיים. תוכנית צריכה להחליף את הסדר של המערך.

תרגיל מס' 37.

כתוב תוכנית שמקבלת מערך בעל 20 מספרים שלמים חיוביים אקראיים. יש לאפס את כל האיברים שנמצאים בין מינימום למקסימום של מערך.

תרגיל מס' 38.

כתוב תוכנית שמקבלת מערך בעל 20 מספרים שלמים חיוביים אקראיים. תוכנית מעתיקה איברים למערך חדש ומאפסת את כל האיברים שיותר גדולים מהשכנים שלהם משמאל ומימין:

1 2 5 5 3 1 6 1 19 4 4 7 3 1 5 3 16 4 1 3 2

תרגיל מס' 39.

כתוב תוכנית שמקבלת מערך בעל 10 מספרים שלמים חיוביים אקראיים וקולטת מהמשתמש את מספר הזזות ימינה. תוכנית מזיזה את המערך ומדפיסה את התוצאה, לדוגמה:

2 5 1 19 4 3 5 16 11 2

N=3

0 0 0 2 5 1 19 4 3 5

תרגיל מס' 40.

כתוב תוכנית שמקבלת מערך בעל 10 מספרים שלמים חיוביים אקראיים וקולטת מהמשתמש את מספר סיבובים שמאלה. תוכנית מסובבת את המערך ומדפיסה את התוצאה, לדוגמה:

2 5 1 19 4 3 5 16 11 2

N=3

19 4 3 5 16 11 2 2 5 1



## פרק ח'. רשימות דו-ממדיות (מטריצות).

### מה זה מטריצה?

בתכנות מטריצות נקראות בדרך כלל מערכים דו-ממדיים. דו ממדי פירושו, כל אלמנט במערך עצמו הוא מערך. מערכים מקוננים אלה הם בדרך כלל חד ממדיים, אחרת מדובר על מטריצות תלת ממדיות או רב ממדיות.

בנוסף, לכל המערכים המקוננים חייבים להיות בעלי אותו אורך זה עם זה, מכיוון שהמטריצה מיוצגת בדרך כלל בצורה של מלבן רגיל, כלומר הטבלה הדו-ממדית הרגילה בה כל השורות שוות באורך.

ממד הטבלה, נמדד במספר השורות והעמודות. תורגם לשפת המטריצה, מספר השורות הוא מספר המערכים המקוננים, מספר העמודות הוא מספר האלמנטים במערך מקונן אחד.

מכיוון שבפייתון "מערך" זה סוג "עני" המובנה בשפה, אבל יש "רשימה" דומה הרבה יותר למערך בשפות אחרות, אנו נבנה מטריצות מהרשימה. נניח שאנחנו צריכים מטריצה 3x4 של מספרים אקראיים.

נראה את הדוגמה למילוי שלה ותצוגה:

```
from random import randint
N = 3
M = 4
a = []
for i in range(N):
    b = []
    for j in range(M):
        b.append(randint(1,99))
    a.append(b)
print(a, end='\n\n')
for i in range(N):
    for j in range(M):
        print("%3d" % a[i][j], end=" ")
print()
```

```
[[7, 83, 51, 74], [49, 66, 98, 9], [44, 65, 19, 26]]
```

```
7 83 51 74
49 66 98 9
44 65 19 26
```

ראשית, אנו מדפיסים את המטריצה בשורה אחת, אך ורק כדי להראות כיצד הנתונים נשמרים ברשימה.

פנייה למרכיבי המטריצה מתרחשת באמצעות האינדקסים שלהם. ראשית, מסומן מספר השורה ואז מספר העמודה. אז כדי לחלץ את המספר 7 בדוגמה שלמעלה, יש לכתוב את הביטוי: `a[1][0]`, מכיוון שהמספר 7 נמצא בשורה השנייה, שהאינדקס שלה הוא 1, ובטור הראשון, שהאינדקס שלו הוא 0.

בפייתון, עקב אפשרויות של הלולאה for אין צורך להשתמש באינדקסים בעת ביצוע ספירה. מחזור הפלט של המסך עשוי להיראות כך:

```
from random import randint
N = 3
M = 4
a = []
for i in range(N):
    b = []
    for j in range(M):
        b.append(randint(1,99))
    a.append(b)
print(a, end='\n\n')

for i in a:
    for j in i:
        print("%3d" % j, end=" ")
    print()
```

```
[[78, 51, 84, 61], [74, 33, 59, 68], [70, 55, 97, 77]]
```

```
78 51 84 61
74 33 59 68
70 55 97 77
```

כאן אנו מוציאים לראשונה את הרשימה הבאה מתוך a. לאחר חזרות על רשימת קינון זה בלולאה מקוננת. כשלומדים תכנות, כשמדובר במטריצות, פותרים המון משימות: מציאת אלמנטים מסוימים, חישוב סכומים, מיון וכדומה. בעקרון, משימות אלה קשורות בספירת אלמנטים בצורה כזו או אחרת. בפייתון תחום זה מפושט יותר בגלל נוכחותם של פונקציות ושיטות מובנות, למרות שאף אחד לא אוסר להשתמש וללמוד אלגוריתמים קלאסיים.

ניתן להגדיר מטריצה ישירות בתוך הקוד בצורה הבאה:

```
A = [[1, 4, 5, 12],
      [-5, 8, 9, 0],
      [-6, 7, 11, 19]]
```

לא חובה להגדיר תת רשימות בשורות חדשות, אפשר גם כך:

```
A = [[1, 4, 5, 12], [-5, 8, 9, 0], [-6, 7, 11, 19]]
```

אם נדפיס שניהם, נקבל אותה תוצאה:

```
print("A =", A)
```

```
A = [[1, 4, 5, 12], [-5, 8, 9, 0], [-6, 7, 11, 19]]
```

כדי לגשת לשורה מסוימת (בעצם לרשימה מסוימת) צריכים לסמן את מספר השורה בסוגריים מרובעות:

```
print("second line =", A[1])
```

```
second line = [-5, 8, 9, 0]
```

כדי לגשת תא או תאים ספציפיים יש לסמן את 2 קואורדינטות שלהם:

```
print("A[1][2] =", A[1][2])  
print("A[0][-1] =", A[0][-1])
```

```
A[1][2] = 9  
A[0][-1] = 12
```

כדי לגשת לאיברים של עמודות אנו צריכים "לטייל" בין רשימות כדי להוציא איבר-איבר.

```
column = []  
for row in A:  
    column.append(row[2])  
print("3rd column =", column)
```

```
3rd column = [5, 9, 11]
```

**דוגמאות:**

מיון.

נניח יש צורך למיין את האיברים בשורות המטריצה. ניתן להשתמש באחד מאלגוריתמי המיון, למשל, במיון בועות:

```
from random import randint  
N = 3  
M = 4  
# פונקציה להדפסת מטריצה  
def output_matrix(matrix):  
    for row in matrix:  
        for item in row:
```

```

print("%3d" % item, end="")
print()
# מילוי מטריצה
a = []
for i in range(N):
    b = []
    for j in range(M):
        b.append(randint(1,99))
    a.append(b)
# הדפסת מטריצה לפני מיון
output_matrix(a)
print()
# מיון בשורות של מטריצה
for i in range(N):
    for j in range(M-1):
        for k in range(M-j-1):
            if a[i][k] > a[i][k+1]:
                a[i][k], a[i][k+1] = a[i][k+1], a[i][k]
# הדפסת מטריצה לאחר מיון
output_matrix(a)

```

```

76 35 83 12
35 70 44 12
66 40 66 6

12 35 76 83
12 35 44 70
6 40 66 66

```

מצד שני לשפת פייתון יש פונקציה מובנת שנקראת `sort()`

```

from random import randint
N = 3
M = 4
# פונקציה להדפסת מטריצה
def output_matrix(matrix):
    for row in matrix:
        for item in row:
            print("%3d" % item, end="")
        print()
# מילוי מטריצה
a = []
for i in range(N):
    b = []

```

```

for j in range(M):
    b.append(randint(1,99))
a.append(b)
# הדפסת מטריצה לפני מיון
output_matrix(a)
print()
# מיון בשורות של מטריצה
for i in range(N):
    a[i].sort()
# הדפסת מטריצה לאחר מיון
output_matrix(a)

```

```

61 45 83 9
27 88 44 84
34 83 34 77

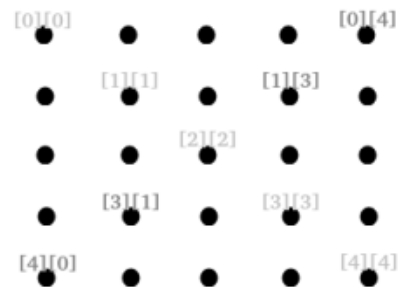
9 45 61 83
27 44 84 88
34 34 77 83

```

מצא סכום של איברים של אלכסון ראשי ואלכסון משני במטריצה ריבועית.

במטריצות מרובעות (עבור כאלה, מספר השורות שווה למספר העמודות) ישנם אלכסונים. האלכסון הראשי עובר מהפינה השמאלית העליונה לפינה הימנית התחתונה. אלכסון משני – מהפינה הימנית העליונה לפינה השמאלית התחתונה. האינדקס הראשון והשני של כל אלמנט באלכסון הראשי זהה. אז האלמנט הראשון הוא  $[0][0]$ , אחריו הוא  $[1][1]$  וכו'.

המדדים של כל אלמנט באלכסון המשני בסך הכול צריכים לתת את הממד של המטריצה מינוס 1 (אם האינדקס הוא מאפס).



```

from random import randint
N = 5
# מייצרים מטריצה ומדפיסים אותה
a = []
for i in range(N):
    b = []
    for j in range(N):

```

```

n = randint(1, 9)
b.append(n)
print("%3d" % n, end="")
a.append(b)
print()
# מוציאים סכום של האלכסון הראשי והמשני
diagonal1 = 0
diagonal2 = 0
for i in range(N):
    diagonal1 += a[i][i]
    diagonal2 += a[i][N-1-i]
print(diagonal1)
print(diagonal2)

```

```

2 3 6 4 2
4 3 4 5 5
5 1 1 8 1
1 4 7 1 3
8 5 3 2 7
14
20

```

## ספרייה מיוחדת לטיפול במטריצות numpy.

קיימת ספרייה עם אוסף פעולות מיוחדות על מטריצות. קודם כל אנו צריכים להתקין את האוסף:

```

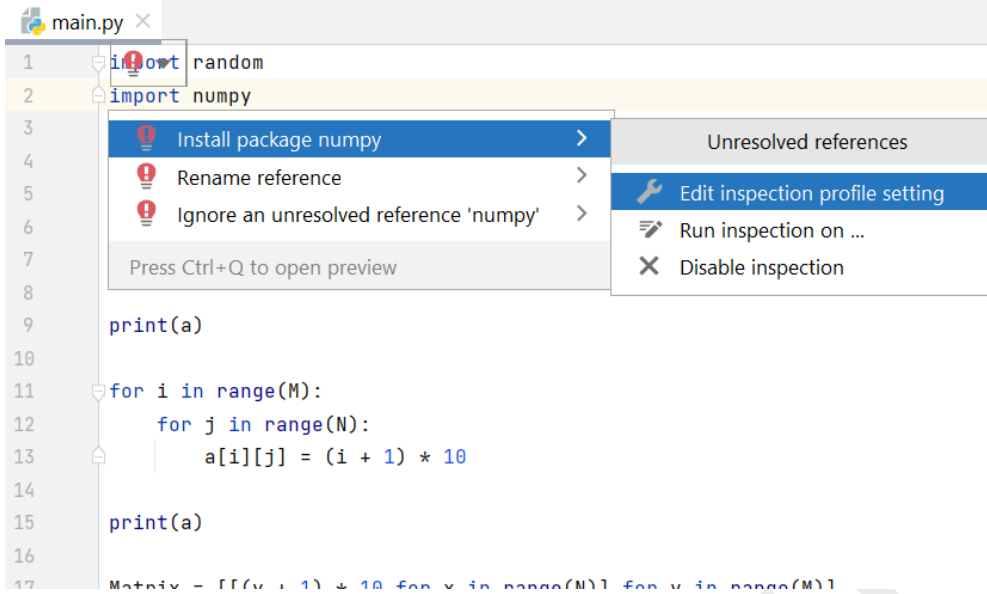
C:\Users\marka>C:\Users\marka\AppData\Local\Programs\Python\Python310\python.exe -m pip install --upgrade pip
Requirement already satisfied: pip in c:\users\marka\appdata\local\programs\python\python310\lib\site-packages (22.0.3)
Collecting pip
  Downloading pip-22.0.4-py3-none-any.whl (2.1 MB)
----- 2.1/2.1 MB 86.3 kB/s eta 0:00:00
Installing collected packages: pip
  Attempting uninstall: pip
    Found existing installation: pip 22.0.3
    Uninstalling pip-22.0.3:
      Successfully uninstalled pip-22.0.3
Successfully installed pip-22.0.4

C:\Users\marka>pip install numpy
Collecting numpy
  Downloading numpy-1.22.3-cp310-cp310-win_amd64.whl (14.7 MB)
----- 14.7/14.7 MB 43.5 kB/s eta 0:00:00
Installing collected packages: numpy
Successfully installed numpy-1.22.3

C:\Users\marka>

```

מתוך PyCharm צריכים "לשחרר" את השגיאה שמודול לא קיים:



לאחר התקנה נחקור את הקוד הפשוט ביותר:

```
import random
import numpy
M = random.randrange(2, 10)
N = random.randrange(2, 10)
print("M = ", M, "; N = ", N)
a = numpy.zeros((M, N))
print(a)
for i in range(M):
    for j in range(N):
        a[i][j] = (i + 1) * 10
print(a)
Matrix = [[(y + 1) * 10 for x in range(N)] for y in range(M)]
print(Matrix)
```

לאחר הרצה נראה תמונה הבאה:

```
M = 4 ; N = 6
[[0. 0. 0. 0. 0. 0.]
 [0. 0. 0. 0. 0. 0.]
 [0. 0. 0. 0. 0. 0.]
 [0. 0. 0. 0. 0. 0.]
 [10. 10. 10. 10. 10. 10.]
 [20. 20. 20. 20. 20. 20.]
 [30. 30. 30. 30. 30. 30.]
 [40. 40. 40. 40. 40. 40.]
 [[10, 10, 10, 10, 10, 10], [20, 20, 20, 20, 20, 20], [30, 30, 30, 30, 30, 30], [40, 40, 40, 40, 40, 40]]
```

רשימת פונקציות לטיפול במטריצות:

פונקציה	הסבר
<b>array()</b>	פונקציה להגדרת המטריצה או מערך
<b>add()</b>	פונקציה מסכמת איברים של 2 מטריצות
<b>subtract()</b>	פונקציה מחסרת איברים של 2 מטריצות
<b>divide()</b>	פונקציה מחלקת איברים של 2 מטריצות
<b>multiply()</b>	פונקציה להכפלת איברים של 2 מטריצות
<b>dot()</b>	פונקציה להכפלת 2 מטריצות
<b>sqrt()</b>	פונקציה לחישוב שורש ריבועי מכל אלמנט של מטריצה
<b>sum(x,axis)</b>	פונקציה לסיכום את כל האיברים של מטריצה. axis - הוא הפרמטר האופציונלי לחישוב סכומים של עמודות (axis=0) או שורות (axis=1).
<b>T</b>	פונקציה להפיכת מטריצה (Transpose).
<b>zeros()</b>	פונקציה ליצירת מטריצת אפסים
<b>ones()</b>	פונקציה ליצירת מטריצת אחדים

דוגמת שימוש בפונקציות numpy:

```
import numpy
# Two matrices are initialized by value
x = numpy.array([[1, 2], [4, 5]])
y = numpy.array([[7, 8], [9, 10]])
# add() is used to add matrices
print ("Addition of two matrices: ")
print (numpy.add(x,y))
# subtract() is used to subtract matrices
print ("Subtraction of two matrices : ")
print (numpy.subtract(x,y))
# divide() is used to divide matrices
print ("Matrix Division : ")
print (numpy.divide(x,y))
print ("Multiplication of two matrices: ")
print (numpy.multiply(x,y))
print ("The product of two matrices : ")
print (numpy.dot(x,y))
print ("square root is : ")
print (numpy.sqrt(x))
print ("The summation of elements : ")
print (numpy.sum(y))
print ("The column wise summation : ")
print (numpy.sum(y,axis=0))
print ("The row wise summation: ")
print (numpy.sum(y,axis=1))
```



```
# using "T" to transpose the matrix
print ("Matrix transposition : ")
print (x.T)
```

Addition of two matrices:

```
[[ 8 10]
 [13 15]]
```

Subtraction of two matrices :

```
[[ -6 -6]
 [ -5 -5]]
```

Matrix Division :

```
[[0.14285714 0.25   ]
 [0.44444444 0.5   ]]
```

Multiplication of two matrices:

```
[[ 7 16]
 [36 50]]
```

The product of two matrices :

```
[[25 28]
 [73 82]]
```

square root is :

```
[[1.      1.41421356]
 [2.      2.23606798]]
```

The summation of elements :

```
34
```

The column wise summation :

```
[16 18]
```

The row wise summation:

```
[15 19]
```

Matrix transposition :

```
[[1 4]
 [2 5]]
```

נבדוק מספר אפשרויות נוספות של ספריית NumPy:

```
import numpy as np
a = np.arange(15).reshape(3, 5)
print("Matrix a ", a)

print("Shape ", a.shape)
print("Dimensions ", a.ndim)
print(a.dtype.name)
print(a.itemsize)
print(a.size)
```

הכנסת מודול numpy וייצירת קיצור np שירשור של 2 הוראות: הוראה ראשונה לבניית מערך חד ממדי ומילוי מ-0 עד 14. הוראה שנייה מארגנת מטריצה דו-ממדית של 3 על 5. הצגת מידות של מטריצה הצגת ממדיים של מטריצה הצגת סוג נתונים שנמצאים בתוך מטריצה הצגת גודל של כל נתון ביחידות B הצגת כמות נתונים במטריצה הצגת סוג נתונים של מטריצה כולה

```
print(type(a))
b = np.linspace( 0, 2, 9 )
print(b)
```

ייצור וקטור b בין 0 ל-2 שמכיל 9 איברים עם מרווחים שווים. הצגץ ווקטור.

נראה את הפלט של הקוד:

```
Matrix a [[ 0  1  2  3  4]
 [ 5  6  7  8  9]
 [10 11 12 13 14]]
Shape (3, 5)
Dimmensions 2
int32
4
15
<class 'numpy.ndarray'>
[0.  0.25 0.5  0.75 1.  1.25 1.5  1.75 2. ]
```

עוד מספר פעולות:

```
import numpy as np
a = np.array( [20,30,40,50] )
b = np.arange( 4 )
print(b)
c = a-b
print(c)
b=b**2
print(b)
a=10*np.sin(a)
print(a)
print( a<5)
```

הכנסת מודול numpy וייצירת קיצור np הגדרת ווקטור עם ערכים מוגדרים מראש הגדרת ווקטור עם ערכי ברירת מחדל 0,1,2,3

חיסור ווקטורי

עליית איברי ווקטור בחזקת 2

מחשבים סינוס של ווקטור, מכפלים את התוצאה ב- 10 (כל איבר) עושים השוואה של כל איבר ל מספר 5

```
[0 1 2 3]
[20 29 38 47]
[0 1 4 9]
[ 9.12945251 -9.88031624  7.4511316 -2.62374854]
[False True False True]
```

עוד דוגמה לפעולות ומילוי מטריצות:

```
import numpy as np
a = np.ones((2,3), dtype=int)
b = np.random.random((2,3))
a *= 3
print(a)
```

מילוי מטריצה 2 על 3 באחדים מסוג שלם יצירת מטריצה 2 על 3 ומילוי בערכים אקראיים הכפלת כל הערכים במטריצה ראשונה ב- 3

```
b += a
print(b)
a += b
```

חיבור מטריצות float=float+int

נקבל שגיאה כי float+int לא יכול להיכנס ל-int

מספר פונקציות נוספות (חישוב סכום מטריצה, מינימום, מקסימום). כפי שאתם רואים יש אופציה לעבוד גם בתור פונקציה רגילה, גם דרך פונקציה של ספריית numpy.

```
import numpy as np
a=(10*np.random.random((2,3)))
print(a)
print("Sum = ",a.sum())
print("Sum = ",np.sum(a))
print("Min = ",a.min())
print("Min = ",np.min(a))
print("Max = ",a.max())
print("Max = ",np.max(a))
```

```
[[8.71226231 3.38910556 4.10284202]
 [8.21574323 3.00488121 5.38358057]]
Sum = 32.808414899268854
Sum = 32.808414899268854
Min = 3.0048812129322675
Min = 3.0048812129322675
Max = 8.712262308749057
Max = 8.712262308749057
```

איחוד מטריצות:

קיימות 2 פונקציות לאיחוד מטריצות: איחוד אופקי ואיחוד אנכי: vstack, hstack.

```
import numpy as np
a = np.floor(10*np.random.random((2,2)))
print(a)
b = np.floor(10*np.random.random((2,2)))
print(b)
print(np.vstack((a,b)))
print(np.hstack((a,b)))
```

```
[[9. 3.]
 [4. 5.]]
[[0. 6.]
 [3. 7.]]
[[9. 3.]
```

```
[4. 5.]
[0. 6.]
[3. 7.]
[[9. 3. 0. 6.]
[4. 5. 3. 7.]]
```

פיצול מטריצות:

```
import numpy as np
a = np.floor(10*np.random.random((2,12)))
print(a)
print(np.hsplit(a,3))
print(np.hsplit(a,(3,4)))
```

```
[[0. 6. 5. 1. 7. 6. 6. 9. 1. 3. 5. 0.]
 [3. 7. 2. 7. 1. 1. 3. 9. 7. 8. 2. 0.]
 [array([[0., 6., 5., 1.],
        [3., 7., 2., 7.]]), array([[7., 6., 6., 9.],
        [1., 1., 3., 9.]]), array([[1., 3., 5., 0.],
        [7., 8., 2., 0.]])]
 [array([[0., 6., 5.],
        [3., 7., 2.]]), array([[1.],
        [7.]])], array([[7., 6., 6., 9., 1., 3., 5., 0.],
        [1., 1., 3., 9., 7., 8., 2., 0.]])]
```

תרגול בנושא רשימות דו-ממדיות.  
תרגול בנושא טבלאות מעקב:  
תרגיל מס' 1.

תפענח את הקוד הבא: מה יהיה פלט של תוכנית?

```
A = [[1, 4, 5, 12],
      [-5, 8, 9, 0],
      [-6, 7, 11, 19]]
print("A =", A)
print("A[1] =", A[1])
print("A[1][2] =", A[1][2])
print("A[0][-1] =", A[0][-1])
column = []
for row in A:
    column.append(row[2])
print("3rd column =", column)
```

תרגיל מס' 2.

תפענח את הקוד הבא: מה יהיה פלט של תוכנית?

```
N = 3
M = 3
res = [ [ i*j for i in range(N) ] for j in range(M) ]
print("Inline representation:")
[ [ print(res[i][j], end = " ") for i in range(N) ] for j in range(M) ]
print("")
print("Multiline representation")
for i in range(N):
    for j in range(M):
        print(res[i][j], end = " ")
    print("")
```

תרגיל מס' 3.

תפענח את הקוד הבא: מה יהיה פלט של תוכנית?

```
N = 3
M = 3
lis=[[0,0,0],[0,0,0],[0,0,0]]
for i in range(N):
    for j in range(M):
        lis[i][j]=i+j
for i in range(N):
    for j in range(M):
        print(lis[i][j],end=" ")
    print("")
```

תרגיל מס' 4.

תפענח את הקוד הבא: מה יהיה פלט של תוכנית?

```
import random
import numpy as np
M = random.randrange(2, 10)
N = random.randrange(2, 10)
print("M = ", M, "; N = ", N)
a = np.zeros((M, N))
for i in range(M):
    for j in range(N):
        a[i][j] = (j + 1) * 5
print(a)
```

```
Matrix = [[(x + 1) * 5 for x in range(N)] for y in range(M)]
print(Matrix)
```

תרגיל מס' 5.

תפענח את הקוד הבא: מה יהיה פלט של תוכנית?

```
import numpy as np
def f(x, y):
    return 10 * x + y
a = np.arange(10)**3
print(a)
print(a[2])
print(a[2:5])
a[:6:2] = -1000
print(a)
print(a[: :-1])
for i in a:
    print(i, end=" ")
print()
b = np.fromfunction(f,(5,4),dtype=int)
print(b)
print(b[2,3])
print(b[0:5, 1])
print(b[: ,1])
print(b[1:3, :])
```

תרגיל מס' 6.

תפענח את הקוד הבא: מה יהיה פלט של תוכנית?

```
M1 = [[8, 14, -6],[12,7,4],[-11,3,21]]
M2 = [[3, 16, -6],[9,7,-4],[-1,3,13]]
M3 = [[0,0,0],[0,0,0],[0,0,0]]
matrix_length = len(M1)
for i in range(len(M1)):
    for k in range(len(M2)):
        M3[i][k] = M1[i][k] + M2[i][k]
print("The answer is ", M3)
print()
for i in range(len(M1)):
    for k in range(len(M2)):
        M3[i][k] = M1[i][k] + M2[k][i]
```

```

print("The answer is ", M3)
print()
for i in range(len(M1)):
    for k in range(len(M2)):
        M3[i][k] = M1[k][i] + M2[i][k]
print("The answer is ", M3)

```

תרגיל מס' 7.

תפענח את הקוד הבא: מה יהיה פלט של תוכנית? תצמצם את הקוד ככל שאפשר, בלי לשנות את אופי של תוכנית.

```

M1 = [[8, 14, -6],[12,7,4],[-11,3,21]]
M2 = [[3, 16, -6],[9,7,-4],[-1,3,13]]
M3 = [[0,0,0],[0,0,0],[0,0,0]]
matrix_length = len(M1)
for i in range(len(M1)):
    for k in range(len(M2)):
        if i==k:
            M3[i][k] = M1[i][k] + M2[i][k]
print("The answer is ", M3)
print()
for i in range(len(M1)):
    for k in range(len(M2)):
        if i!=k:
            M3[i][k] = M1[i][k] - M2[k][i]
        else:
            M3[i][k]=0
print("The answer is ", M3)

```

תרגיל מס' 8.

תפענח את הקוד הבא: מה יהיה פלט של תוכנית?

```

import random
import numpy

M = random.randrange(2,10)
N = random.randrange(2,10)
print("M = ",M,"; N = ",N)
a = numpy.zeros((M, N))

c = [random.randrange(1,5) for i in range(M)]
print("Set of ",M," numbers:")
print(c)

```

```

for i in range(M):
    for j in range(N):
        a[i][j] = c[i]
print(a)

Matrix = []
for i in range(M):
    Matrix.append([c[i] for j in range(N)])
print(Matrix)

```

תרגיל מס' 9.

תפענח את הקוד הבא: מה יהיה פלט של תוכנית?

```

import random
import numpy
M = random.randrange(2,10)
N = random.randrange(2,10)
print("M = ",M,"; N = ",N)
a = numpy.zeros((M, N))
c = [random.randrange(1,5) for i in range(N)]
print("Set ",N," of nums:")
print(c)
print()
for i in range(M):
    for j in range(N):
        a[i][j] = c[j]
print(a)
print()
Matrix = [c for y in range(M)]
print(Matrix)

```

תרגיל מס' 10.

תפענח את הקוד הבא: מה יהיה פלט של תוכנית?

```

import random
import numpy

M = random.randrange(2,10)
N = random.randrange(2,10)
Q = random.randrange(-5,5)
print("M = ",M,"; N = ",N,"; Q = ",Q)
a = numpy.zeros((M, N))

```



```

c = [random.randrange(1,5) for i in range(N)]
print("Set ",N," of Nums:")
print(c)
print()

for i in range(M):
    for j in range(N):
        a[i][j] = c[j]*Q**i
print(a)
print()

Matrix = []
for i in range(M):
    Matrix.append([c[j]*Q**i for j in range(N)])
print(Matrix)

```

תרגיל מס' 11.

תפענח את הקוד הבא: מה יהיה פלט של תוכנית?

```

import random
import numpy
M = random.randrange(2,10)
N = random.randrange(2,10)
K = random.randrange(0,M)
print("M = ",M,"; N = ",N,"; K = ",K)
a = numpy.zeros((M, N))
for i in range(M):
    for j in range(N):
        a[i][j] = random.randrange(-5,5)
print(a)
print("Row ",K," : ")
print(a[K])
print()

Matrix = [[random.randrange(-5,5) for x in range(N)] for y in range(M)]
print(Matrix)
print("Row ",K," : ")
print(Matrix[K])

```

תרגיל מס' 12.

תפענח את הקוד הבא: מה יהיה פלט של תוכנית?

```
import random
import numpy
M = random.randrange(2,10)
N = random.randrange(2,10)
K = random.randrange(0,N)
print("M = ",M,"; N = ",N,"; K = ",K)
a = numpy.zeros((M, N))
for i in range(M):
    for j in range(N):
        a[i][j] = random.randrange(-5,5)
print(a)
print("Column ",K," : ")
for i in range(M):
    print(a[i][K],end="; ")
print()
Matrix = [[random.randrange(-5,5) for x in range(N)] for y in range(M)]
print(Matrix)
print("Column ",K," : ")
for i in range(M):
    print(Matrix[i][K],end="; ")
print()
```

תרגיל מס' 13.

תפענח את הקוד הבא: מה יהיה פלט של תוכנית?

```
import random
import numpy
M = random.randrange(2,10)
N = random.randrange(2,10)
print("M = ",M,"; N = ",N)
a = numpy.zeros((M, N))
for i in range(M):
    for j in range(N):
        a[i][j] = random.randrange(-5,5)
print(a)
print("Even Rows: ")
for i in range(0,M,2):
    print(a[i])
print()
Matrix = [[random.randrange(-5,5) for x in range(N)] for y in range(M)]
```

```

print(Matrix)
print("Even Rows: ")
for i in range(0,M,2):
    print(Matrix[i])
print()

```

תרגיל מס' 14.

תפענח את הקוד הבא: מה יהיה פלט של תוכנית?

```

import random
import numpy
M = random.randrange(2,10)
N = random.randrange(2,10)
print("M = ",M,"; N = ",N)
a = numpy.zeros((M, N))
for i in range(M):
    for j in range(N):
        a[i][j] = random.randrange(-5,5)
print(a)
print("Odd Columns: ")
for i in range(M):
    print(a[i][0::2])
print()
Matrix = [[random.randrange(-5,5) for x in range(N)] for y in range(M)]
print(Matrix)
print("Odd Columns: ")
for i in range(M):
    print(Matrix[i][0::2])
print()

```

תרגיל מס' 15.

תפענח את הקוד הבא: מה יהיה פלט של תוכנית?

```

import random
import numpy
M = random.randrange(2,10)
N = random.randrange(2,10)
print("M = ",M,"; N = ",N)
a = numpy.zeros((M, N))
k = 0
for i in range(M):
    for j in range(N):
        k += 1

```

```

    a[i][j] = k
print(a)
for i in range(M):
    if i%2 == 0:
        print(a[i])
    else:
        print(a[i][::-1])
b = []
for i in range(M):
    b.append([])
    for j in range(N):
        b[i].append(a[i][j])
print(b)
for i in range(M):
    if i%2 == 0:
        print(b[i])
    else:
        print(b[i][::-1])

```

תרגיל מס' 16.

תפענח את הקוד הבא: מה יהיה פלט של תוכנית?

```

import random
import numpy
M = random.randrange(2,10)
N = random.randrange(2,10)
print("M = ",M,"; N = ",N)
a = numpy.zeros((M, N))
k = 0
for i in range(M):
    for j in range(N):
        k += 1
        a[i][j] = k
print(a)
for i in range(M):
    for j in range(N):
        if j%2 == 0:
            print(a[i][j],end=" ")
        else:
            print(a[M-1-i][j],end=" ")
print()

```

תרגיל מס' 17.

תפענח את הקוד הבא: מה יהיה פלט של תוכנית?

```
import numpy
M = 5
print("M = ",M)
a = numpy.zeros((M, M))
k = 0
for i in range(M):
    for j in range(M-i):
        k += 1
        a[i][j] = k
        k += M-i-1
k = M+1
for j in range(1,M):
    for i in range(j,M):
        a[i][M-j] = k
        k += 1
        k += M-j
print(a)
for i in range(M):
    for j in range(M-i):
        print(a[i][j], end=" ")
    print()
    for j in range(i+1,M):
        print(a[j][M-i-1], end=" ")
```

תרגיל מס' 18.

תפענח את הקוד הבא: מה יהיה פלט של תוכנית?

```
import numpy
M = 4
print("M = ",M)
a = numpy.zeros((M, M))
k = 0
for i in range(M):
    for j in range(M-i):
        k += 1
        a[j][i] = k
        k += M-i-1
k = M+1
for j in range(1,M):
    for i in range(j,M):
```

```

    a[M-j][i] = k
    k += 1
    k += M-j
print(a)
for i in range(M):
    for j in range(M-i):
        print(a[j][i], end=" ")
    print()
    for j in range(i+1,M):
        print(a[M-i-1][j], end=" ")

```

תרגיל מס' 19.

תפענח את הקוד הבא: מה יהיה פלט של תוכנית?

```

import numpy
M = 5
print("M = ",M)
a = numpy.zeros((M, M))
k = 0
for i in range(0,M):
    for j in range(i,M-i):
        k += 1
        a[i][j] = k
    for j in range(i+1,M-i):
        k += 1
        a[j][M-1-i] = k
    for j in range(M-2-i,i-1,-1):
        k += 1
        a[M-1-i][j] = k
    for j in range(M-2-i,i,-1):
        k += 1
        a[j][i] = k
    i +=2
print(a)
k = 0
for i in range(0,M):
    for j in range(i,M-i):
        print(a[i][j], end=" ")
    print()
    for j in range(i+1,M-i):
        print(a[j][M-1-i], end=" ")
    print()
    for j in range(M-2-i,i-1,-1):
        print(a[M-1-i][j], end=" ")
    print()

```

```
for j in range(M-2-i,i,-1):
    print(a[j][i], end=" ")
i +=2
```

תרגיל מס' 20.

תפענח את הקוד הבא: מה יהיה פלט של תוכנית?

```
import random
import numpy as np
M = random.randrange(2,7)
N = random.randrange(2,7)
K = random.randrange(0,M)
print("M = ",M,"; N = ",N,"; K = ",K)
a = np.zeros((M, N))
for i in range(M):
    for j in range(N):
        a[i][j] = random.randrange(1,5)
print(a)
Sum = sum(a[K])
Product = np.product(a[K])
print(a[K])
print("Sum: ",Sum)
print("Product: ",Product)
```

תרגיל מס' 21.

תפענח את הקוד הבא: מה יהיה פלט של תוכנית?

```
import random
import numpy as np
M = random.randrange(2,7)
N = random.randrange(2,7)
K = random.randrange(0,N)
print("M = ",M,"; N = ",N,"; K = ",K)
a = np.zeros((M, N))
for i in range(M):
    for j in range(N):
        a[i][j] = random.randrange(1,5)
print(a)
b = a.sum(axis=1)
print(b)
```

תרגיל מס' 22.

תפענח את הקוד הבא: מה יהיה פלט של תוכנית?

```
import random
import numpy as np
M = random.randrange(2,7)
N = random.randrange(2,7)
K = random.randrange(0,N)
print("M = ",M,"; N = ",N)
a = np.zeros((M, N))
for i in range(M):
    for j in range(N):
        a[i][j] = random.randrange(1,5)
print(a)
b = a.prod(0)
print(b)
```

תרגיל מס' 23.

תפענח את הקוד הבא: מה יהיה פלט של תוכנית?

```
import random
import numpy as np
M = random.randrange(2,8)
N = random.randrange(2,8)
print("M = ",M,"; N = ",N)
a = np.random.randint(5, size=(M, N))
print(a)
b = a[1::2]
print()
print(b)
for x in b:
    print("Row:",x)
    print("Mean:", np.mean(x))
```

תרגיל מס' 24.

תפענח את הקוד הבא: מה יהיה פלט של תוכנית?

```
import random
import numpy as np
M = random.randrange(2,8)
N = random.randrange(2,8)
print("M = ",M,"; N = ",N)
```



```

a = np.random.randint(5, size=(M, N))
print(a)
b = a[:,0::2]
print()
print("Odd columns:")
print(b)
print(b.mean(axis=0))
print()
b = b.transpose()
for x in b:
    print("Column:",x)
    print("Mean:",np.mean(x))

```

תרגיל מס' 25.

תפענח את הקוד הבא: מה יהיה פלט של תוכנית?

```

import random
import numpy as np
M = random.randrange(2,8)
N = random.randrange(2,8)
print("M = ",M,"; N = ",N)
a = np.random.randint(15, size=(M, N))
print(a)
print()
for x in a:
    print("Row:",x)
    print("Minimum:",min(x))

```

תרגיל מס' 26.

תפענח את הקוד הבא: מה יהיה פלט של תוכנית?

```

import random
import numpy as np
M = random.randrange(2,8)
N = random.randrange(2,8)
print("M = ",M,"; N = ",N)
a = np.random.randint(5, size=(M, N))
print(a)
print()
for j in range(N):
    x = a[:,j]
    print("Column:",x)
    print("Maximum:",max(x))

```

תרגיל מס' 27.

תפענח את הקוד הבא: מה יהיה פלט של תוכנית?

```
import random
import numpy as np
M = random.randrange(2,8)
N = random.randrange(2,8)
print("M = ",M,"; N = ",N)
a = np.random.randint(5, size=(M, N))
print(a)
print()
S = []
for x in a:
    S.append(sum(x))
    print("Row:",x)
print()
print(S)
S_max = max(S)
print("Maximum of sums:",S_max)
idx = [i for i, x in enumerate(S) if x == S_max]
print("Place of Max:",idx)
```

תרגיל מס' 28.

תפענח את הקוד הבא: מה יהיה פלט של תוכנית? מה קורה אם יש יותר ממקסימום 1.

```
import random
import numpy as np
M = random.randrange(2,8)
N = random.randrange(2,8)
print("M = ",M,"; N = ",N)
a = np.random.randint(5, size=(M, N))
print(a)
S = np.prod(a,axis=0)
print("\nProduct of columns:")
print(S)
S_max = max(S)
print("\nMaximum of products:",S_max)
idx = [i for i, x in enumerate(S) if x == S_max]
print("Place of Max:",idx)
```

תרגיל מס' 29.

תפענח את הקוד הבא: מה יהיה פלט של תוכנית?

```
import random
import numpy as np
M = random.randrange(2,8)
N = random.randrange(2,8)
print("M = ",M,"; N = ",N)
a = np.random.randint(5, size=(M, N))
print(a)
print()
mins = []
for x in a:
    print("Row:",x)
    print("Minimum:",min(x))
    mins.append(min(x))
print("\nMinimums:")
print(mins)
print("Max of Mins:",max(mins))
```

תרגיל מס' 30.

תפענח את הקוד הבא: מה יהיה פלט של תוכנית?

```
import random
import numpy as np
M = random.randrange(2,8)
N = random.randrange(2,8)
print("M = ",M,"; N = ",N)
a = np.random.randint(5, size=(M, N))
print(a)
print()
maxs = []
for j in range(N):
    x = a[:,j]
    print("Column:",x)
    print("Maximum:",max(x))
    maxs.append(max(x))
print("\nMaximums:")
print(maxs)
print("Minimum of Maxs:",min(maxs))
```

תרגיל מס' 31.

תפענח את הקוד הבא: מה יהיה פלט של תוכנית?

```
import random
import numpy as np
M = random.randrange(2,8)
N = random.randrange(2,8)
print("M = ",M,"; N = ",N)
a = np.random.randint(5, size=(M, N))
print(a)
print()
for x in a:
    x_mean = np.mean(x)
    c = sum(item < x_mean for item in x)
    print("Row:",x)
    print("Mean:",x_mean)
    print("Number of items < Mean:",c)
    print()
```

תרגיל מס' 32.

תפענח את הקוד הבא: מה יהיה פלט של תוכנית?

```
M = random.randrange(2,8)
N = random.randrange(2,8)
print("M = ",M,"; N = ",N)
a = np.random.randint(5, size=(M, N))
print(a)
print()
for j in range(N):
    x = a[:,j]
    x_mean = np.mean(x)
    c = sum(item > x_mean for item in x)
    print("Column:",x)
    print("Mean:",x_mean)
    print("Number of items > Mean:",c)
    print()
```

תרגיל מס' 33.

תפענח את הקוד הבא: מה יהיה פלט של תוכנית?

```
from random import randint
N = 5
M = 7
matrix = []
for i in range(N):
    row = []
    for j in range(M):
        row.append(randint(0, 20))
    matrix.append(row)
for row in matrix:
    for item in row:
        print("%4d" % item, end="")
    print()
low = int(input("Lower limit: "))
high = int(input("Upper limit: "))
count = 0
for i in range(N):
    for j in range(M):
        if low <= matrix[i][j] <= high:
            count += 1
print("Number of elements is:", count)
```

תרגיל מס' 34.

תפענח את הקוד הבא: מה יהיה פלט של תוכנית?

```
from random import randint
N = 5
M = 7
matrix = []
for i in range(N):
    row = []
    for j in range(M):
        row.append(randint(0, 20))
    matrix.append(row)
for row in matrix:
    for item in row:
        print("%4d" % item, end="")
    print()
num = int(input("Number to find: "))
for i in range(N):
```

```

for j in range(M):
    if num == matrix[i][j]:
        print("[{0}, {1}].format(i, j),end=" ")

```

תרגיל מס' 35.

תפענח את הקוד הבא: מה יהיה פלט של תוכנית?

```

from random import random
M = 10
N = 5
a = []
for i in range(N):
    b = []
    for j in range(M):
        b.append(round(random()*10))
    a.append(b)
    print(b)
c1 = int(input("Source column: ")) - 1
c2 = int(input("Destination column: ")) - 1
for i in range(N):
    a[i][c1], a[i][c2] = a[i][c2], a[i][c1]
    print(a[i])

```

תרגיל מס' 36.

תפענח את הקוד הבא: מה יהיה פלט של תוכנית?

```

from random import random
M = 10
N = 5
a = []
for i in range(N):
    b = []
    for j in range(M):
        b.append(int(random()*11))
        print("%3d" % b[j], end="")
    a.append(b)
    print(' |', sum(b))
for i in range(M):
    print(" --", end="")
print()
for i in range(M):

```

```

s = 0
for j in range(N):
    s += a[j][i]
print("%3d" % s, end="")
print()

```

תרגיל מס' 37.

תפענח את הקוד הבא: מה יהיה פלט של תוכנית?

```

from random import randint
N = 5
M = 7
matrix = []
for i in range(N):
    row = []
    for j in range(M):
        row.append(randint(0, 20))
    matrix.append(row)
for row in matrix:
    for item in row:
        print("%4d" % item, end="")
    print()
low = int(input("Lower limit: "))
high = int(input("Upper limit: "))
count = 0
for row in matrix:
    for item in row:
        if low <= item <= high:
            count += 1
print("Number of elements is:", count)

```

תרגיל מס' 38.

תפענח את הקוד הבא: מה יהיה פלט של תוכנית?

```

import random
n = 4
m = 6
matrix = []
for i in range(n):
    line = []
    for j in range(m):
        line.append(random.randint(0, 5))
    matrix.append(line)

```

```

for line in matrix:
    for i in line:
        print("%4d" % i, end="")
    print()
print()
for line in matrix:
    summa = 0
    i = 0
    while i < m - 1:
        summa += line[i]
        i += 1
    else:
        line[m - 1] = summa
for line in matrix:
    for i in line:
        print("%4d" % i, end="")
    print()

```

תרגיל מס' 39.

תפענח את הקוד הבא: מה יהיה פלט של תוכנית?

```

from random import random
M = 10
N = 5
a = []
for i in range(N):
    b = []
    for j in range(M):
        b.append(int(random()*11))
        print("%3d" % b[j], end="")
    a.append(b)
    print()
for i in range(M):
    print(" --", end="")
print()
max_sum = 0
col = 0
for i in range(M):
    s = 0
    for j in range(N):
        s += a[j][i]
    print("%3d" % s, end="")
    if s > max_sum:
        max_sum = s

```



```
col = i
print()
print(col+1)
```

תרגיל מס' 40.

תפענח את הקוד הבא: מה יהיה פלט של תוכנית?

```
from random import random
N = 5
matrix = []
for i in range(N):
    row = []
    for j in range(N):
        row.append(int(random()*10))
    matrix.append(row)
for row in matrix:
    print(row)
sumMain = 0
sumSecondary = 0
for i in range(N):
    sumMain += matrix[i][i]
    sumSecondary += matrix[i][N-i-1]
print(sumMain)
print(sumSecondary)
```

תרגיל מס' 41.

תפענח את הקוד הבא: מה יהיה פלט של תוכנית?

```
from random import random
N = 5
a = []
for i in range(N):
    z = []
    for j in range(N):
        n = int(random() * 100)
        z.append(n)
        print("%4d" % n, end="")
    print()
    a.append(z)
print()
for i in range(N):
    for j in range(N):
        if i==j:
            b = a[i][j]
```

```

a[i][j] = a[i][N-1-j]
a[i][N-1-j] = b
for i in range(N):
    for j in range(N):
        print("%4d" % a[i][j], end="")
    print()

```

תרגיל מס' 42.

תפענח את הקוד הבא: מה יהיה פלט של תוכנית?

```

from random import random
M = 10
N = 5
a = []
for i in range(N):
    b = []
    for j in range(M):
        n = int(random()*200)
        b.append(n)
        print("%4d" % n, end="")
    a.append(b)
    print()
print("Minimum are")
mx = -1
for j in range(M):
    mn = 200
    for i in range(N):
        if a[i][j] < mn:
            mn = a[i][j]
        print(mn, end=" ")
    if mn > mx:
        mx = mn
print()
print("Maximum of minimum: ", mx)

```

תרגיל מס' 43.

תפענח את הקוד הבא: מה יהיה פלט של תוכנית?

```

from random import random
N = 5
M = 7
matrix = []
for i in range(N):
    a = [0] * M

```

```

for j in range(M):
    a[j] = int(random()*10)
    print("%3d" % a[j],end=")
matrix.append(a)
print()
suma = [0] * M
for j in range(M):
    for i in range(N):
        suma[j] += matrix[i][j]
    print("%3d" % suma[j],end=")
print("\n")
for j in range(M-1):
    for i in range(M-j-1):
        if suma[i] > suma[i+1]:
            suma[i],suma[i+1] = suma[i+1],suma[i]
            for k in range(N):
                matrix[k][i],matrix[k][i+1] = matrix[k][i+1],matrix[k][i]
for i in matrix:
    for j in i:
        print("%3d"%j,end=")
    print()
for i in suma:
    print("%3d"%i,end=")
print()

```

## תרגול בנושא כתיבת קוד.

תרגיל מס' 1.

הגדר רשימה דו-ממדית (מטריצה) בגודל  $M$  על  $N$  בעזרת גנרטור של מספר אקראי. תמלא אותה כך שמספר שנמצא בתוך המטריצה הוא סכום המקומות לפי שורה ועמודה, לדוגמה איבר שעומד בשורה 2 ועמודה 3 מכיל ערך 5. הדפס את המטריצה לאחר מילוי.

תרגיל מס' 2.

א. הגדר רשימה דו-ממדית (מטריצה) בגודל  $M$  על  $N$  בעזרת גנרטור של מספר אקראי. תמלא אותה עם מספר רץ מ-0 עד  $M*N-1$ . הדפס את המטריצה לאחר מילוי. לדוגמה:

0 1 2 3 4 5 6

7 8 9 10 11 12 13

14 15 16 17 18 19 20

ב. הדפס מטריצה בצורה יפה כך, שכל מספר יהיה בעמודה משלו. שים לב שיכול להיות מקרה של מספר חד ספרתי, דו ספרתי ותלת ספרתי.

0 1 2 3 4 5 6

7 8 9 10 11 12 13

14 15 16 17 18 19 20

תרגיל מס' 3.

הגדר רשימה דו-ממדית (מטריצה) בגודל  $M$  על  $N$  בעזרת גנרטור של מספר אקראי. תמלא אותה כך ששורה עליונה תכיל מספר עמודה, עמודה ימנית תכיל מספר שורה וכל היתר איברים יהיו מאופסים.

תרגיל מס' 4.

הגדר רשימה דו-ממדית (מטריצה) בגודל  $M$  על  $N$  בעזרת גנרטור של מספר אקראי. תמלא אותה כך ששורה עליונה ושורה תחתונה תכילו מספר עמודה, עמודה ימנית ועמודה שמאלית תכילו מספר שורה וריבוע פנימי יהיה מאופס.

תרגיל מס' 5.

הגדר רשימה דו-ממדית (מטריצה) בגודל  $M$  על  $N$  בעזרת גנרטור של מספר אקראי וספרה  $D$  שנקלטת מהמשתמש (בין 1 ל-9). תמלא אותה כך ששורה עליונה ושורה תחתונה תכילו מספר עמודה, עמודה ימנית ועמודה שמאלית תכילו מספר שורה וריבוע פנימי מורכב מכפל של מספר שורה במספר עמודה במידה והוא לא מתחלק ב- $D$  ללא שארית או 0 במקרה והוא מתחלק ב- $D$  ללא שארית.

תרגיל מס' 6.

כתוב תוכנית שמגדירה 2 רשימות דו-ממדיות (מטריצות) בגודל  $M$  על  $N$  בעזרת גנרטור של מספר אקראי וממלאה אותם בערכים אקראיים בין 0 ל-10.

תוכנית צריכה לסכם 2 מטריצות האלה ומדפיסה גם מטריצות מקור וגם תוצאת החיבור.

תרגיל מס' 7.

בדוק כמה איברים ברשימה דו-ממדית נמצאים בטווח מסוים.

זהו חיפוש פשוט בתוך מטריצה. בלולאה אנו צריכים לעבור על שורות. בלולאה הפנימית, עברו על כל שורה ובדקו כל איבר האם הוא נמצא בטווח. אם הוא כלול בתוכו, יש להגדיל את המונה של אלמנטים כאלה ב-1.

תרגיל מס' 8.

נתונה רשימה דו-ממדית. קולטים מספר מהמשתמש. התוכנית צריכה להחזיר כתובת תאים שבהם נמצא מספר זה.

תרגיל מס' 9.

נתונה רשימה דו-ממדית  $10 \times 5$  יש להחליף איברים של עמודה  $x$  עם עמודה  $y$ .

העמודה של המטריצה מוגדרת על ידי האינדקס השני. לכן, בלולאה, יש צורך במיון בשורות המטריצה, כלומר משתנה של המונה ישמש כאינדקס הראשון של המערך הדו-ממדי.

בכל שורה, החלף את האלמנטים עם המספרים של העמודות שצוינו. ניתן לבצע החלפה באמצעות משתנה שלישי.

אפשר להגדיר את מספרי העמודות שיש להחליף על ידי הזנת ערכים בזמן בו התוכנית פועלת.

אם בשפת התכנות הצמדת המערכים מתחילה מאפס, והעמודות מסומנות על סמך מיקומן בפועל במטריקס החל מ-1, אז זה נובע מהמשתנים שבהם נכתבים מספרי העמודות, נוריד אחד לפני השימוש בלולאה.

תרגיל מס' 10.

יש לחשב סכום של כל עמודה (יש להדפיס אותה מתחת כל עמודה) ושל כל שורה (יש להדפיס אותה מימין של כל שורה).

מכיוון שמערך דו-ממדי בדרך כלל נבדק לפי שורות, קל יותר לחשב את הסכום של כל שורה.

אפשר למלא שורה של מטריצה ולהציג את האלמנטים שלה על המסך, לצבור את סכום האלמנטים בשורה במשתנה ולהציג אותו בסוף השורה.

בפייתון ניתן לחשב את סכומי שורות המטריצה באמצעות הפונקציה  $\text{sum}()$  המועברת לרשימה הנוכחית - מחרוזת.

סכומי העמודות מחושבים על ידי חציית כל עמודה במטריצה. שימו לב שההיפך הוא הנכון כאן: הלולאה החיצונית חוצה את העמודים, הלולאה הפנימית חוצה את השורות.

תרגיל מס' 11.

יש להחליף איבר אחרון של עמודה בסכום של כל האיברים של השורה. אלגוריתם לפתרון החלק העיקרי של הבעיה (רק חיבור והחלפה):

1. לכל שורה, הקצה את הערך 0 למשתנה  $\text{summa}$ , האינדקס של האלמנט  $i$  גם לערך 0.
2. עד שמדד האלמנט יגיע לאלמנט האחרון בשורה, הגדל את ערך הסכום בערך האלמנט הבא בשורת המטריצה.
3. אחרת (כאשר אינדקס האלמנטים מצביע על האלמנט האחרון במחרוזת), שנה את הערך של האלמנט האחרון במחרוזת לערך של משתנה  $\text{summa}$ .

תרגיל מס' 12.

נתונה רשימה דו-ממדית שמורכבת ממספרים חיוביים בלבד. יש לחשב את סכום האיברים בכל עמודה ולמצוא איזה עמודה מכילה סכום גדול ביותר.

לאחר שנחשב את סכום האלמנטים בעמודה הבאה, השווה אותה לערך המשתנה המיועד לאחסון הסכום המקסימלי. אם הנוכחי מתגלה כגדול יותר, אנו כותבים אותו למשתנה שצוין. בנוסף, נזכור את מספר העמודה הנוכחית במשתנה נפרד.

אם המטריצה מכילה מספר עמודות עם אותו סכום מקסימלי וצריך לקבוע את כולם, אז הבעיה נפתרת קצת אחרת. במקרה זה, יש לזכור את סכומי העמודות לצורך השוואה בהמשך למקסימום שנמצא בעבר.

תרגיל מס' 13.

מצא סכום של האלכסון הראשי והאלכסון המשני של רשימה דו-ממדית ריבועית.

רק למטריצה ריבועית יש אלכסונים. מטריצה ריבועית היא מטריצה עם מספר השורות ששווה למספר העמודות. האלכסון הראשי עובר מהפינה השמאלית העליונה לימין התחתון, האלכסון המשני מהפינה הימנית העליונה לשמאל התחתון.

מרכיבי האלכסון הראשי עם אותו אינדקס גם לפי ציר  $X$  וגם לפי ציר  $Y$ . עבור אלמנטים באלכסון המשני, האינדקס השני מנוגד לראשון.

## תרגיל מס' 14.

החלף איברים של האלכסון הראשי והאלכסון המשני ביניהם של רשימה דו-ממדית ריבועית. רק למטריצה ריבועית יש אלכסונים. מטריצה ריבועית היא מטריצה עם מספר השורות שווה למספר העמודות. האלכסון הראשי עובר מהפינה השמאלית העליונה לימין התחתון, האלכסון המשני מהפינה הימנית העליונה לשמאל התחתון.

מרכיבי האלכסון הראשי עם אותו אינדקס גם לפי ציר  $X$  וגם לפי ציר  $Y$ . עבור אלמנטים באלכסון המשני, האינדקס השני מנוגד לראשון.

מכיוון שהחלפה מתבצעת בשורה אחת, לערכים שיש להחליף יש את המדדים הבאים:

- עבור אלמנט באלכסון הראשי - המדדים הראשון והשני שווים זה לזה ( $i = j$ ),
- עבור אלמנט באלכסון המשני - האינדקס הראשון הוא  $i$ , והשני הוא  $N-1-j$  (כאשר אינדקס מתחיל מאפס).
- כאשר עוברים דרך המטריצה, חילופי הערכים מתרחשים רק במקרה בו  $i = j$

## תרגיל מס' 15.

בכל עמודה של מטריצה דו-ממדית יש למצוא את האיבר הקטן ביותר. יש למצוא מקסימום בין כל האיברים הקטנים ביותר שמצאתם.

משימה זו מורכבת משתי משימות משנה:

1) מצא את האלמנט המינימלי שלך בכל טור במטריצה

2) השווה בין הנמוכים שנמצאו ובחר את המקסימום מהם. באחד המשתנים נשמור את האלמנט המינימלי בעמודה הנוכחית. לאחר הצגת העמודה, נשווה את ערכה לערך המשתנה המסופק לאחסון האלמנט המרבי בין המינימום. הערכים הראשוניים של משתנים אלה צריכים להיות: לאחסן את המינימום הנוכחי - ערך גדול יותר מהמקסימום האפשרי, לאחסן את המקסימום בין המינימום, ערך נמוך מהמינימום האפשרי.

לדוגמה, אם טווח המטריצה הוא בין 0 ל 199, יש להקצות משתנה אחד 200, והשני - 1.

מכיוון שהעמודים עוברים איטרציה, ערכו של המשתנה האחראי לשורות משתנה בלולאה הפנימית. בעת הצגת מטריצה על המסך, אנו מבצעים את ההסקה החדה הרגילה.

## תרגיל מס' 16.

ביצוע מיון של עמודות של מטריצה דו-ממדית לפי סכום האיברים שלהם.

כשמוצאים את סכום האלמנטים בעמודה מסוימת, האינדקס הראשון משתנה, כלומר מספר השורה. בתוכנית שלהלן, הסכומים שנמצאו של אלמנטים בעמודה מוזנים לרשימה חד ממדית. מספר האלמנטים ברשימה זו שווה למספר העמודות במטריקס. בשלב הבא, רשימת הכמויות ממוינת לפי מיון בועות. אם הערכים ברשימת הסכומים מוחלפים, יש להחליף את העמודות המתאימות במטריצה. מדדי הערכים שהוחלפו ברשימת הסכומים זהים למדדי העמודות שהוחלפו. כשמחליפים ערכי עמודות, יש לעבור בין כל שורות המטריצה. כך, לצד מיון רשימה חד ממדית, העמודות של המטריצה ממוינות. בסוף התוכנית, שוב מופיעים סכומי המטריצה והעמודה, אך הם ממוינים.

תרגיל מס' 17.

כתוב תוכנית שמייצרת מטריצה  $5 \times 3$ . האיברים מקבלים ערכים אקראיים מ-10 עד 10. בכל מקום שיש איבר שלילי תוצג "-". מצא את הסכום של כל שורה ושל כל עמודה של איברים חיוביים בלבד.

תרגיל מס' 18.

הגדר רשימה דו-ממדית (מטריצה) בגודל  $M$  על  $N$  בעזרת גנרטור של מספר אקראי. כתוב תוכנית שמעתיקה את המטריצה המקורית למטריצה סימטרית לפי ציר  $X$ :

0	1	2	3	4	5	6	14	12	10	11	18	10	50
7	8	<u>9</u>	<u>10</u>	11	12	13	1	8	<u>0</u>	11	15	13	
14	15	16	17	18	19	20	0	1	5	3	4	2	0

בסופו של תהליך תוכנית מדפיסה גם מטריצת מקור וגם את התוצאה.

תרגיל מס' 19.

הגדר רשימה דו-ממדית (מטריצה) בגודל  $M$  על  $N$  בעזרת גנרטור של מספר אקראי. כתוב תוכנית שמעתיקה את המטריצה המקורית למטריצה סימטרית לפי ציר  $Y$ :

0	1	2	3	4	5	6	0	2	4	4	5	1	0
7	8	<u>9</u>	<u>10</u>	11	12	13	41	51	11	<u>01</u>	0	8	7
14	15	16	17	18	19	20	05	01	81	71	01	21	41

בסופו של תהליך תוכנית מדפיסה גם מטריצת מקור וגם את התוצאה.

תרגיל מס' 19.

הגדר רשימה דו-ממדית (מטריצה) בגודל  $M$  על  $N$  בעזרת גנרטור של מספר אקראי. כתוב תוכנית שמעתיקה את המטריצה המקורית למטריצה מסובבת ב-180 מעלות:

0	1	2	3	4	5	6	14	15	16	17	18	19	20
7	8	<u>9</u>	<u>10</u>	11	12	13	41	21	11	<u>01</u>	6	8	7
14	15	16	17	18	19	20	9	5	4	3	2	1	0

בסופו של תהליך תוכנית מדפיסה גם מטריצת מקור וגם את התוצאה.



תרגיל מס' 20.

הגדר רשימה דו-ממדית (מטריצה) בגודל M על N בעזרת גנרטור של מספר אקראי. כתוב תוכנית שמעתיקה את המטריצה המקורית למטריצה מסובבת ב- 90 מעלות עם כיוון השעון:

0	1	2	3	4	5	6	0	1	2	3	4	5	6
7	8	<u>9</u>	<u>10</u>	11	12	13	7	8	<u>9</u>	<u>10</u>	11	12	13
14	15	16	17	18	19	20	14	15	16	17	18	19	20

בסופו של תהליך תוכנית מדפיסה גם מטריצת מקור וגם את התוצאה.

תרגיל מס' 21.

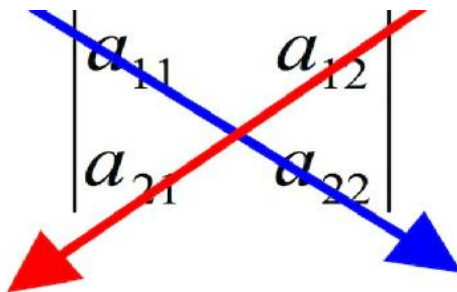
הגדר רשימה דו-ממדית (מטריצה) בגודל M על N בעזרת גנרטור של מספר אקראי. כתוב תוכנית שמעתיקה את המטריצה המקורית למטריצה מסובבת ב- 90 מעלות נגד כיוון השעון:

0	1	2	3	4	5	6	0	1	2	3	4	5	6
7	8	<u>9</u>	<u>10</u>	11	12	13	7	8	9	<u>10</u>	11	12	13
14	15	16	17	18	19	20	14	15	16	17	18	19	20

בסופו של תהליך תוכנית מדפיסה גם מטריצת מקור וגם את התוצאה.

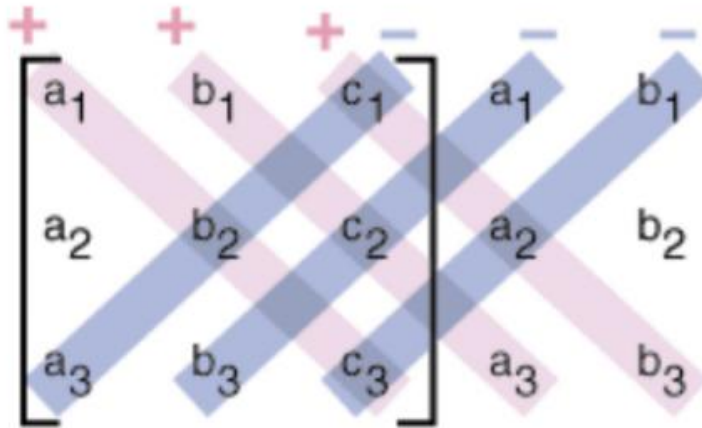
תרגיל מס' 22.

הגדר רשימה דו-ממדית (מטריצה) בגודל 2 על 2 וחשב דטרמיננטה שלה:

$$\begin{vmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{vmatrix} = a_{11} \cdot a_{22} - a_{12} \cdot a_{21}$$


תרגיל מס' 23.

הגדר רשימה דו-ממדית (מטריצה) בגודל 3 על 3 וחשב דטרמיננטה שלה:



$$\det A = (a_1 b_2 c_3 + b_1 c_2 a_3 + c_1 a_2 b_3) - (a_3 b_2 c_1 + b_3 c_2 a_1 + c_3 a_2 b_1)$$

תרגיל מס' 24.

הגדר מטריצה 2 על 3. תקלוט נתוני מטריצה מהמשתמש. כתוב פונקציה שמחשבת דטרמיננטה 2 על 2. פתור מערכת משוואות - 2 משוואות עם 2 נעלמים בעזרת שיטת קראמר:

$$\left. \begin{array}{l} a_1 x + b_1 y = c_1 \\ a_2 x + b_2 y = c_2 \end{array} \right\} \quad D_x = \begin{vmatrix} c_1 & b_1 \\ c_2 & b_2 \end{vmatrix} = c_1 \cdot b_2 - c_2 \cdot b_1$$

$$D = \begin{vmatrix} a_1 & b_1 \\ a_2 & b_2 \end{vmatrix} = a_1 \cdot b_2 - a_2 \cdot b_1, \quad D_y = \begin{vmatrix} a_1 & c_1 \\ a_2 & c_2 \end{vmatrix} = a_1 \cdot c_2 - a_2 \cdot c_1$$

במידה ו- D שונה מ-0 פתרון הוא:

$$x = \frac{D_x}{D}, y = \frac{D_y}{D}$$

אם D שווה ל-0 יש 2 אפשרויות:

אם גם D<sub>x</sub> וגם D<sub>y</sub> שווים ל-0 - אין סוף פתרונות; אחרת - אין פתרון.

בדוק תוכנית בכל אחד מ-3 אפשרויות:

$$\begin{pmatrix} 1 & 2 & 3 \\ 2 & 4 & 6 \end{pmatrix}, \begin{pmatrix} 1 & 2 & 3 \\ 2 & 4 & 5 \end{pmatrix}, \begin{pmatrix} 1 & 2 & 3 \\ 3 & 4 & 2 \end{pmatrix}$$

תרגיל מס' 25.

הגדר מטריצה 3 על 4. תקלוט נתוני מטריצה מהמשתמש. כתוב פונקציה שמחשבת דטרמיננטה 3 על 3. פתור מערכת משוואות - 3 משוואות עם 3 נעלמים בעזרת שיטת קראמר:

$$\left. \begin{aligned} a_1x + b_1y + c_1z &= d_1 \\ a_2x + b_2y + c_2z &= d_2 \\ a_3x + b_3y + c_3z &= d_3 \end{aligned} \right\}$$

$$D = \begin{vmatrix} a_1 & b_1 & c_1 \\ a_2 & b_2 & c_2 \\ a_3 & b_3 & c_3 \end{vmatrix}, \quad D_x = \begin{vmatrix} d_1 & b_1 & c_1 \\ d_2 & b_2 & c_2 \\ d_3 & b_3 & c_3 \end{vmatrix},$$

$$D_y = \begin{vmatrix} a_1 & d_1 & c_1 \\ a_2 & d_2 & c_2 \\ a_3 & d_3 & c_3 \end{vmatrix}, \quad D_z = \begin{vmatrix} a_1 & b_1 & d_1 \\ a_2 & b_2 & d_2 \\ a_3 & b_3 & d_3 \end{vmatrix},$$

במידה ו-D שונה מ-0 פתרון הוא:

$$x = \frac{D_x}{D}, y = \frac{D_y}{D}, z = \frac{D_z}{D}$$

אם D שווה ל-0 יש 2 אפשרויות:

אם גם Dx וגם Dy וגם Dz שווים ל-0 - אין סוף פתרונות; אחרת - אין פתרון. בדוק תוכנית בכל אחד מ-3 אפשרויות:

$$\left( \begin{array}{cccc|c} 2 & 1 & -1 & 1 & 8 \\ 1 & 2 & 1 & 1 & 1 \\ 3 & -1 & -1 & -2 & -1 \end{array} \right), \left( \begin{array}{cccc|c} 2 & 4 & 2 & 2 & 1 \\ 1 & 2 & 1 & 1 & 1 \\ 3 & -1 & -1 & -2 & -1 \end{array} \right), \left( \begin{array}{cccc|c} 2 & 4 & 2 & 2 & -1 \\ 1 & 2 & 1 & 1 & -1 \\ 3 & -1 & -1 & -2 & -1 \end{array} \right)$$

תרגיל מס' 26.

הגדר מטריצה 2 על 3. תקלוט נתוני מטריצה מהמשתמש. פתור מערכת משוואות - 2 משוואות עם 2 נעלמים בעזרת שיטת גאוס.

נניח נתונה מערכת משוואות:

$$E1 \quad 2x - 3y = 5$$

$$E2 \quad 3x + 4y = -5$$

נרשום אותה בצורת מטריצה:

$$\begin{array}{l} R1 \\ R2 \end{array} \left( \begin{array}{cc|c} 2 & -3 & 5 \\ 3 & 4 & -5 \end{array} \right)$$

נכפיל משוואה ראשונה ב-3, משוואה שנייה ב-2 ונחסר אותם. תוצאת החיסור מכניסים במקום שורה שנייה.

$$2R_2 - 3R_1 \rightarrow R_2 \left( \begin{array}{cc|c} 2 & -3 & 5 \\ 0 & 17 & -25 \end{array} \right)$$

נכפיל משוואה ראשונה ב-17, שורה שנייה ב-3 ונחסר אותם. תוצאת החיסור מכניסים במקום שורה ראשונה.

$$17R_1 - (-3)R_2 \rightarrow R_1 \left( \begin{array}{cc|c} 34 & 0 & 10 \\ 0 & 17 & -25 \end{array} \right)$$

נחלק שורה ראשונה ב-34 ונקבל X, שורה שנייה ב-17 ונקבל Y.

$$\left( \begin{array}{cc|c} 1 & 0 & \frac{5}{17} \\ 0 & 1 & -\frac{25}{17} \end{array} \right)$$

## פרק ט'. פונקציות מיוחדות של מחרוזות.

המונחים "Upper case" ו"Lower Case" מגיעים מהאופן בו התארגנו בתי דפוס לפני מאות שנים. חלקים נפרדים מסוג מתכת הוחזקו בארגזים שנקראו "cases". האותיות הקטנות, בהן נעשה שימוש לרוב, נשמרו בארגזים ליד העובדים שהיו קלות יותר להגיע אליהם. אותיות גדולות (capital letters), ששימשו בתדירות נמוכה יותר, נשמרו מעל והייה יותר קשה לקחת אותם. בגלל אמנת האחסון הישנה הזו, אנו עדיין מתייחסים לאותיות קטנות כאל "Lower Case" ואותיות גדולות בתור "Upper case".



נגדיר מחרוזת כמו שעשינו בשיעורים ראשונים ונראה כמה פעולות מעניינות:

```
>>> s="Hello World"
>>> s[0]
'H'
>>> s[7:]
'orld'
>>> s[::2]
'HloWrld'
>>> s[::3]
'HIWI'
>>> s[-1]
'd'
>>> s[1]="w"
Traceback (most recent call last):
  File "<pyshell#54>", line 1, in <module>
    s[1]="w"
TypeError: 'str' object does not support item assignment
>>> s="qwerty"
>>> s
'qwerty'
```

## רשימת פונקציות לטיפול במחרוזות.

שימו לב שבהבדל מרשימות אי-אפשר לשנות תו בתוך המחרוזת, אבל כן אפשר לשנות את כל המחרוזת. אפשרות אחרת היא בניית מחרוזת חדשה באותו שם כמו המחרוזת ישנה.

```
>>> s[1]="w"
Traceback (most recent call last):
  File "<pyshell#54>", line 1, in <module>
    s[1]="w"
TypeError: 'str' object does not support item assignment
>>> s="qwerty"
>>> s
'qwerty'
>>> s="Hello World."
>>> s=s[0:-1]+'!!!!'
>>> s
'Hello World!!!!'
```

בפייתון יש הרבה פונקציות טיפול במחרוזות כדי לראות אותם נכתוב פקודה: `dir(str)` ונקבל:

```
>>> dir(str)
['_add_', '__class__', '__contains__', '__delattr__', '__dir__', '__doc__', '__eq__', '__format__',
'__ge__', '__getattr__', '__getitem__', '__getnewargs__', '__gt__', '__hash__', '__init__',
'__init_subclass__', '__iter__', '__le__', '__len__', '__lt__', '__mod__', '__mul__', '__ne__',
'__new__', '__reduce__', '__reduce_ex__', '__repr__', '__rmod__', '__rmul__', '__setattr__',
'__sizeof__', '__str__', '__subclasshook__', 'capitalize', 'casefold', 'center', 'count', 'encode',
'endswith', 'expandtabs', 'find', 'format', 'format_map', 'index', 'isalnum', 'isalpha', 'isascii',
'isdecimal', 'isdigit', 'isidentifier', 'islower', 'isnumeric', 'isprintable', 'isspace', 'istitle', 'isupper',
'join', 'ljust', 'lower', 'lstrip', 'maketrans', 'partition', 'replace', 'rfind', 'rindex', 'rjust', 'rpartition',
'rsplit', 'rstrip', 'split', 'splitlines', 'startswith', 'strip', 'swapcase', 'title', 'translate', 'upper', 'zfill']
```

כדי לקבל עזרה כיצד משתמשים בפונקציה ספציפית נכתוב:

```
>>> help(str.split)
Help on method_descriptor:

split(self, /, sep=None, maxsplit=-1)
    Return a list of the words in the string, using sep as the delimiter string.

    sep
    The delimiter according which to split the string.
    None (the default value) means split according to any whitespace,
    and discard empty strings from the result.
    maxsplit
    Maximum number of splits to do.
    -1 (the default value) means no limit.
```

## פונקציות לטיפול במחרוזות עם הסברים ו דוגמאות.

פונקציה שמחלקת את המחרוזת לחלקים. בברירת מחדל, כאשר הפונקציה לא מקבלת אף פרמטר היא מחלקת את המחרוזת לפי מילים, כאשר סימן רווח הוא סימן שמפריד בין המילים. ניתן לתת לפונקציה פרמטר וזה יהיה תו שהפונקציה תחפש ותפריד את המשפט בהתאם לתווים האלה.	split()
---	---------

```
>>> s="Haifa TelAviv Karmiel Eilat Galil"
>>> s
'Haifa TelAviv Karmiel Eilat Galil'
>>> s1=s.split()
>>> s1
['Haifa', 'TelAviv', 'Karmiel', 'Eilat', 'Galil']
>>> s2=s.split('a')
>>> s2
['H', 'if', ' TelAviv K', 'rmiel Eil', 't G', 'lil']
```

פונקציה הפוכה מהפונקציה split() – היא מאחדת את המילים למחרוזת. כמו גם אפשר לקבוע את הפרמטר של איחוד מילים. שימו לב שחייבים לתת פרמטר שאותו צריכים להכניס בין המילים. המחרוזת עצמה נשארת ללא שינוי – אפשר או להדפיס אותה או להעתיק אותה למחרוזת אחרת.	join()
--	--------

```
>>> join(s1)
Traceback (most recent call last):
  File "<pyshell#75>", line 1, in <module>
    join(s1)
NameError: name 'join' is not defined
>>> ".join(s1)
'Haifa TelAviv Karmiel Eilat Galil'
>>> .join(s2)
SyntaxError: invalid syntax
>>> "e".join(s2)
'Heife TelAviv Kermiel Eilet Gelil'
>>> s2
['H', 'if', ' TelAviv K', 'rmiel Eil', 't G', 'lil']
>>> s= "a".join(s2)
>>> s
'Haifa TelAviv Karmiel Eilat Galil'
```

<p>פונקציה מחפשת תת מחרוזת (או תו בודד) בתוך מחרוזת גדולה ומחזירה את המיקום של <b>ההופעה הראשונה של תת המחרוזת במחרוזת הגדולה</b>. אם היא מוצאת אותה, היא מחזירה את מיקומה, אם לא היא מחזירה -1. פונקציה עובדת באחד מ-3 אפשרויות: עם פרמטר אחד, עם 2 פרמטרים, עם 3 פרמטרים. פרמטר ראשון –תת-המחרוזת או האות שהיא מחפשת. פרמטר שני – מאיזה מקום במחרוזת היא מחפשת. פרמטר שלישי – עד איזה מיקום במחרוזת היא מחפשת. אם אין פרמטר שלישי – היא מחפשת מפרמטר שני ועד הסוף, אם פרמטר שני ושלישי לא נמצאים – אז היא מחפשת מהתחלה ועד הסוף.</p>	find()
--	--------

```
>>> s
'Haifa TelAviv Karmiel Eilat Galil'
>>> s.find("a")
1
>>> s.find("a",2)
4
>>> s.find("a",7,9)
-1
```

<p>הפונקציה מחליפה את תת המחרוזת המקורית לתת מחרוזת נבחרת. רק בפעם ראשונה שהיא מופיעה במחרוזת הגדולה. המחרוזת עצמה לא משתנה. אם רוצים לשמור את התוצאה – יש להעתיק אותה למחרוזת חדשה.</p>	replace()
--	-----------

```
>>> s
'Haifa TelAviv Karmiel Eilat Galil'
>>> s.replace("Haifa", "Jerusalem")
'Jerusalem TelAviv Karmiel Eilat Galil'
>>> s
'Haifa TelAviv Karmiel Eilat Galil'
>>> t=s.replace("Haifa", "Jerusalem")
>>> s
'Haifa TelAviv Karmiel Eilat Galil'
>>> t
'Jerusalem TelAviv Karmiel Eilat Galil'
```

<p>ראינו את הפקודה בהתחלה, כאשר דיברנו על הדפסה. לפקודה הזאת יש אפשרויות נוספות. במידה ובסוגריים {} אין מספרים, ארגומנטים יהיו מסודרים לפי סדר ההכנסה. אפשר גם למספר אותם איך שבא לנו, ע"י הכנסת מספרים לתוך הסוגריים. ניתן גם לכתוב מילת מפתח בתוך הסוגריים ולתת ערכים לפי מילים האלה.</p>	format()
---	----------

```
>>> s1="length - {}, width - {}, height - {}"
>>> s1.format(10,20,30)
'length - 10 , width - 20, height - 30'
>>> s1="length - {2}, width - {0}, height - {1}"
```



```

>>> s1.format(10,20,30)
'length - 30 , width - 10, height - 20'
>>> s2="Marks of math is {math}, physics is {phy}, english is {eng}"
>>> s2
'Marks of math is {math}, physics is {phy}, english is {eng}'
>>> s2.format(eng="55", math="66", phy="77")
'Marks of math is 66, physics is 77, english is 55'
>>> s2
'Marks of math is {math}, physics is {phy}, english is {eng}'
>>> s2=s2.format(eng="55", math="66", phy="77")
>>> s2
'Marks of math is 66, physics is 77, english is 55'

```

upper()

הפונקציה הופכת את כל האותיות הקטנות לגדולות, אבל לא משנה את המחרוזת המקורית.

```

>>> s2="jsfLJL<34534_+1"
>>> s2
'jsfLJL<34534_+1'
>>> s2.upper()
'JSFLJL<34534_+1'
>>> s1=s2.upper()

```

lower()

הפונקציה הופכת את כל האותיות הגדולות לקטנות, אבל לא משנה את המחרוזת המקורית.

```

>>> s2.lower()
'jsfljl<34534_+1'

```

isupper()

הפונקציה בודקת האם כל האותיות במחרוזת הן אותיות גדולות. אם כן מחזירה TRUE אם לא מחזירה FALSE. הפונקציה לא לוקחת בחשבון ספרות או סימנים מיוחדים.

```

>>> s1.isupper()
True

```

islower()

הפונקציה בודקת האם כל האותיות במחרוזת הן אותיות קטנות. אם כן מחזירה TRUE אם לא מחזירה FALSE. הפונקציה לא לוקחת בחשבון ספרות או סימנים מיוחדים.

```

>>> s1.islower()
False

```

פונקציה בודקת האם **כל התווים** במחרוזת הם ספרות.

isdigit()

```
>>> s3.isdigit()
True
>>> s2.isdigit()
False
```

פונקציה בודקת האם כל תווים במחרוזת הם אותיות. אם כן מחזירה TRUE אם לא מחזירה FALSE.

isalpha()

```
>>> s2="qwer erwer errg er gefg233434"
>>> s4="sejzjLLHLH"
>>> s2.isalpha()
False
>>> s4.isalpha()
True
```

## תרגול במחרוזות:

תרגול בנושא טבלאות מעקב.

תרגיל מס' 1.

תפענח את הקוד הבא. תן דוגמאות לפלט במקרה שקלט הוא תו בודד ובמקרה של מספר תווים.

```
while True:
    C = input('Please enter only one character: ')
    if len(C) == 1:
        break
    print()
print(ord(C))
```

תרגיל מס' 2.

תפענח את הקוד הבא. תן מספר דוגמאות לפלט בהתאם לגנרטור של מספר אקראי.

```
import random
N = random.randrange(32, 126)
c = chr(N)
c_prev = chr(ord(c)-1)
c_next = chr(ord(c)+1)

print("Char: ",c)
print("Char prev.: ",c_prev)
print("Char next: ",c_next)
```

תרגיל מס' 3.

תפענח את הקוד הבא. תן מספר דוגמאות לפלט בהתאם לגנרטור של מספר אקראי.

```
import random
N = random.randrange(1, 27)
print("N = ",N)
for i in range(N):
    print(chr(ord('A')+i), end=" ")
```

תרגיל מס' 4.

תפענח את הקוד הבא. תן מספר דוגמאות לפלט בהתאם לגנרטור של מספר אקראי.

```
import random
N = random.randrange(1, 27)
print("N = ",N)
for i in range(N):
```

```
print(chr(ord('z')-i), end=" ")
```

תרגיל מס' 5.

תפענח את הקוד הבא. תן מספר דוגמאות לפלט בהתאם לגנרטור של מספר אקראי.

```
import random
ru_letters = u"абвгдеёзийклмнопрстуфхъыьэАБВГДЕЁЗИЙКЛМНОПРСТУФХЪЫЬЭ"
en_letters = u"abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ"
heb_letters = u"אבגדהוזחטיכלמנסעפצקרשת"
digits = u"0123456789"
letters = ru_letters + en_letters + heb_letters + digits
C = random.choice(letters)
print("Symbol: ",C)
if ru_letters.find(C) != -1:
    print("Russian")
elif en_letters.find(C) != -1:
    print("Latin")
elif heb_letters.find(C) != -1:
    print("Hebrew")
elif digits.find(C) != -1:
    print("Digit")
```

תרגיל מס' 6.

תפענח את הקוד הבא. מה יהיה פלט, אם קלט יהיה "We will rock You".

```
import random
a=input("Enter the sentence: ")
print(a)
print(a[::-1])
```

תרגיל מס' 7.

תפענח את הקוד הבא. תן מספר דוגמאות לפלט בהתאם לגנרטור של מספר אקראי.

```
import random
ru_letters = u"абвгдеёзийклмнопрстуфхъыьэАБВГДЕЁЗИЙКЛМНОПРСТУФХЪЫЬЭ"
en_letters = u"abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ"
heb_letters = u"אבגדהוזחטיכלמנסעפצקרשת"
digits = u"0123456789"
letters = ru_letters + en_letters + heb_letters + digits
N = random.randrange(1,15)
print("N: ",N)
C = random.choice(letters)
print("Symbol: ",C)
```

```
s = str(C)*N
print("String: ",s)
```

תרגיל מס' 8.

תפענח את הקוד הבא. תן מספר דוגמאות לפלט בהתאם לגנרטור של מספר אקראי.

```
import random
ru_letters = u"абвгдеёзийклмнопрстуфхъыьэАБВГДЕЁЗИЙКЛМНОПРСТУФХЪЫЬЭ"
en_letters = u"abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ"
heb_letters = u"אבגדהוזחטיכלמנסעפצקרשת"
digits = u"0123456789"
letters = ru_letters + en_letters + heb_letters + digits
C1, C2 = random.sample(letters, 2)
print("Symbol 1: ",C1)
print("Symbol 2: ",C2)
N = random.randrange(2, 21, 2)
print("N: ",N)
s = (C1+C2)*int(N/2)
print("String: ",s)
```

תרגיל מס' 9.

תפענח את הקוד הבא. מה יהיה פלט במקרה שקלט הוא "We are the champions".

```
s = input("Enter the string:")
print(s)
print(s[::-1])
```

תרגיל מס' 10.

תפענח את הקוד הבא. תן מספר דוגמאות לפלט בהתאם לגנרטור של מספר אקראי.

```
import random
N = random.randrange(1,6)
s = "BINGO. It is test"
print("String:",s)
print("N:",N)
x = "*" * N
print("Insert:",x)
print(x.join(s))
```

תרגיל מס' 11.

תפענח את הקוד הבא. תן מספר דוגמאות לפלט בהתאם לגנרטור של מספר אקראי.

```

import random
import string
ru_letters = u"абвгдеёзийклмнопрстуфхъыьэАБВГДЕЁЗИЙКЛМНОПРСТУФХЪЫЬЭ"
en_letters = u"abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ"
heb_letters = u"אבגדהוזחטיכלמנסעפצקרשת "
N = random.randrange(1,10)
s_ru = random.sample(ru_letters, N)
print("RU:",s_ru)
N = random.randrange(1,10)
s_en = random.sample(en_letters, N)
print("EN:",s_en)
N = random.randrange(1,10)
s_heb = random.sample(heb_letters, N)
print("HEB:",s_heb)
N = random.randrange(1,10)
s_digit = random.sample(string.digits, N)
print("Digits:",s_digit)
s = s_ru + s_en + s_heb + s_digit + [' ','']
print("Random:",s)
random.shuffle(s)
s = "".join(s)
print(s)
digits = sum(c.isdigit() for c in s)
print("Number of Digits:",digits)

```

תרגיל מס' 12.

תפענח את הקוד הבא. תן מספר דוגמאות לפלט בהתאם לגנרטור של מספר אקראי.

```

import random
import string
ru_letters = u"абвгдеёзийклмнопрстуфхъыьэАБВГДЕЁЗИЙКЛМНОПРСТУФХЪЫЬЭ"
en_letters = u"abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ"
heb_letters = u"אבגדהוזחטיכלמנסעפצקרשת "
en_Upper = u"ABCDEFGHIJKLMNOPQRSTUVWXYZ"

N = random.randrange(1,10)
s_ru = random.sample(ru_letters, N)
print("RU:",s_ru)

N = random.randrange(1,10)
s_en = random.sample(en_letters, N)
print("EN:",s_en)

N = random.randrange(1,10)

```

```

s_heb = random.sample(heb_letters, N)
print("HEB:",s_heb)

N = random.randrange(1,10)
s_digit = random.sample(string.digits, N)
print("Digits:",s_digit)

s = s_ru + s_en + s_heb + s_digit + [' ','']
print("Random:",s)
random.shuffle(s)
s = "".join(s)
print(s)

k = 0
for c in s:
    if c in en_Upper:
        k += 1
print("Number of Upper Latin letters:",k)

```

תרגיל מס' 13.

תפענח את הקוד הבא. תן מספר דוגמאות לפלט בהתאם לגנרטור של מספר אקראי.

```

import random
import string
ru_letters = u"абвгдеёзийклмнопрстуфхъыьэАБВГДЕЁЗИЙКЛМНОПРСТУФХЪЫЬЭ"
en_letters = u"abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ"
heb_letters = u"אבגדהוזחטיכלמנסעפצקרשת "

ru_Lower = u"абвгдеёзийклмнопрстуфхъыьэ"
en_Lower = u"abcdefghijklmnopqrstuvwxyz"

N = random.randrange(1,10)
s_ru = random.sample(ru_letters, N)
print("RU:",s_ru)

N = random.randrange(1,10)
s_en = random.sample(en_letters, N)
print("EN:",s_en)

N = random.randrange(1,10)
s_heb = random.sample(heb_letters, N)
print("HEB:",s_heb)

N = random.randrange(1,10)
s_digit = random.sample(string.digits, N)
print("Digits:",s_digit)

```

```

s = s_ru + s_en + s_heb + s_digit + [';',']
print("Random:",s)
random.shuffle(s)
s = "".join(s)
print(s)

k = 0
for c in s:
    if (c in en_Lower) or (c in ru_Lower):
        k += 1
print("Number of Lower Latin and Cyrillic letters:",k)

```

תרגיל מס' 14.

תפענח את הקוד הבא. תן מספר דוגמאות לפלט בהתאם לגנרטור של מספר אקראי.

```

import random
import string
ru_letters = u"абвгдеёзийклмнопрстуфхъыьэАБВГДЕЁЗИЙКЛМНОПРСТУФХЪЫЬЭ"
en_letters = u"abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ"
heb_letters = u"אבגדהוזחטיכלמנסעפצקרשת "
en_Upper = u"ABCDEFGHIJKLMNOPQRSTUVWXYZ"
N = random.randrange(1,10)
s_ru = random.sample(ru_letters, N)
print("RU:",s_ru)
N = random.randrange(1,10)
s_en = random.sample(en_letters, N)
print("EN:",s_en)
N = random.randrange(1,10)
s_heb = random.sample(heb_letters, N)
print("HEB:",s_heb)
N = random.randrange(1,10)
s_digit = random.sample(string.digits, N)
print("Digits:",s_digit)
s = s_ru + s_en + s_heb + s_digit + [';',']
print("Random:",s)
random.shuffle(s)
s = "".join(s)
s_new = ""
for c in s:
    if c in en_Upper:
        c = c.lower()
    s_new += c
print(s_new)

```



תפענח את הקוד הבא. תן מספר דוגמאות לפלט בהתאם לגנרטור של מספר אקראי.

```
import random
import string
ru_letters = u"абвгдеёзийклмнопрстуфхъыьэАБВГДЕЁЗИЙКЛМНОПРСТУФХЪЫЬЭ"
en_letters = u"abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ"
heb_letters = u"אבגדהוזחטיכלמנסעפצקרשת "
ru_Lower = u"абвгдеёзийклмнопрстуфхъыьэ"
en_Lower = u"abcdefghijklmnopqrstuvwxyz"
N = random.randrange(1,10)
s_ru = random.sample(ru_letters, N)
print("RU:",s_ru)
N = random.randrange(1,10)
s_en = random.sample(en_letters, N)
print("EN:",s_en)
N = random.randrange(1,10)
s_heb = random.sample(heb_letters, N)
print("HEB:",s_heb)
N = random.randrange(1,10)
s_digit = random.sample(string.digits, N)
print("Digits:",s_digit)
s = s_ru + s_en + s_heb + s_digit + [',',']']
print("Random:",s)
random.shuffle(s)
s = "".join(s)
s_new = ""
for c in s:
    if (c in en_Lower) or (c in ru_Lower):
        c = c.upper()
    s_new += c
print(s_new)
```

תפענח את הקוד הבא. תן מספר דוגמאות לפלט בהתאם לגנרטור של מספר אקראי.

```
import random
import string
ru_letters = u"абвгдеёзийклмнопрстуфхъыьэАБВГДЕЁЗИЙКЛМНОПРСТУФХЪЫЬЭ"
en_letters = u"abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ"
heb_letters = u"אבגדהוזחטיכלמנסעפצקרשת "
ru_Lower = u"абвгдежзийклмнопрстуфхцчшщъыьэюя"
en_Lower = u"abcdefghijklmnopqrstuvwxyz"
```

```

ru_Upper = u"АБВГДЕЖЗИЙКЛМНОПРСТУФХЦЧШЩЪЫЬЭЮЯ"
en_Upper = u"ABCDEFGHIJKLMNOPQRSTUVWXYZ"
N = random.randrange(1,10)
s_ru = random.sample(ru_letters, N)
print("RU:",s_ru)
N = random.randrange(1,10)
s_en = random.sample(en_letters, N)
print("EN:",s_en)
N = random.randrange(1,10)
s_heb = random.sample(heb_letters, N)
print("HEB:",s_heb)
N = random.randrange(1,10)
s_digit = random.sample(string.digits, N)
print("Digits:",s_digit)
s = s_ru + s_en + s_heb + s_digit + [' ','']
random.shuffle(s)
s = "".join(s)
print(s)
s_new = ""
for c in s:
    if (c in en_Lower) or (c in ru_Lower):
        c = c.upper()
    elif (c in en_Upper) or (c in ru_Upper):
        c = c.lower()
    s_new += c
print(s_new)

```

תרגיל מס' 17.

תפענח את הקוד הבא. תן דוגמאות לכל פלט האפשרי במקרה של קלט שונה.

```

s = input("Enter the string: ")
print(s)
if '!' in s:
    try:
        x = float(s)
        print("2. float:",x)
    except:
        print("0. Can't convert to float")
else:
    try:
        x = int(s)
        print("1. int:",x)
    except:
        print("0. Can't convert to int")

```

תרגיל מס' 18.

תפענח את הקוד הבא. תן מספר דוגמאות לפלט בהתאם לגנרטור של מספר אקראי.

```
import random
N = random.randrange(100000,1000000)
print("N = ",N)
s = list(str(N))
print(s)
```

תרגיל מס' 19.

תפענח את הקוד הבא. תן מספר דוגמאות לפלט בהתאם לגנרטור של מספר אקראי.

```
import random
N = random.randrange(100000,1000000)
print("N = ",N)
s = list(str(N))
print(s[::-1])
```

תרגיל מס' 20.

תפענח את הקוד הבא. תן מספר דוגמאות לפלט בהתאם לגנרטור של מספר אקראי.

```
import random
import string
N = random.randrange(4,11)
s = ".join(random.choice(string.digits) for x in range(N))
print("String:",s)
L = list(s)
print("List of char-s:",L)
L_int = list(map(int,L))
print("List of int-s:",L_int)
print("Sum of digits:",sum(L_int))
```

תרגיל מס' 21.

תפענח את הקוד הבא. תן מספר דוגמאות לפלט בהתאם לגנרטור של מספר אקראי.

```
import random
import string
import numpy as np
def RndExpr():
    N = random.randrange(2,11)
    lst_digit = list(np.random.choice(list(string.digits), N))
```

```

lst_oper = list(np.random.choice(['+', '-'], N-1))
result = [None]*(2*N - 1)
result[::2] = lst_digit
result[1::2] = lst_oper
return "".join(result)
for i in range(0,5):
    s = RndExpr()
    print()
    print(s, "=", eval(s))
    oper = s[1::2]
    j = 0
    result = int(s[0])
    for x in oper:
        digit = int(s[2*j+2])
        if x == '+':
            result += digit
        else:
            result -= digit
        print(s[2*j+2], x, result)
        j += 1
    print(oper)
    print(s, "=", result)

```

תרגיל מס' 22.

תפענח את הקוד הבא. תן מספר דוגמאות לפלט בהתאם לגנרטור של מספר אקראי.

```

import random
def RndBinary():
    N = random.randrange(1,11)
    return "".join(random.choice('01') for _ in range(N))
for i in range(0,5):
    s_binary = RndBinary()
    print()
    print(s_binary, "=", int(s_binary, 2))
    decimal = 0
    for digit in s_binary:
        decimal = decimal*2 + int(digit)
    print(s_binary, "=", decimal)

```

תרגיל מס' 23.

תפענח את הקוד הבא. תן מספר דוגמאות לפלט בהתאם לגנרטור של מספר אקראי.

```
import random
def int2bin(i):
    if i == 0:
        return "0"
    s = ""
    while i:
        if i & 1 == 1:
            s = "1" + s
        else:
            s = "0" + s
        i = int(i/2)
    return s
for i in range(0,5):
    s_integer = str(random.randrange(1,100))
    i_integer = int(s_integer)
    print('{0} = {1:b}'.format(s_integer,i_integer))
    print('{0} = {1}'.format(s_integer,int2bin(i_integer)))
    print()
```

תרגיל מס' 24.

תפענח את הקוד הבא. תן מספר דוגמאות לפלט בהתאם לגנרטור של מספר אקראי.

```
import random
ru_letters =
u"абвгдежзийклмнопрстуфхцчшщъыьэюяАБВГДЕЖЗИЙКЛМНОПРСТУФХЦЧШЩЪЫЬЭЮЯ"
en_letters = u"abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ"
heb_letters = u"אבגדהוזחטיכלמנסעפצקרשת "
digits = u"0123456789"
letters = ru_letters + en_letters + heb_letters + digits
K1 = random.randrange(1,21)
N1 = random.randrange(1,K1+1)
S1 = ".join(random.choice(letters) for _ in range(K1))
K2 = random.randrange(1,21)
N2 = random.randrange(1,K2+1)
S2 = ".join(random.choice(letters) for _ in range(K2))

print("N1:",N1)
print("K1:",K1)
print("S1:",S1)
```

```

S1_length = len(S1)
print("Length of S1:",S1_length)

print()
print("N2:",N2)
print("K2:",K2)
print("S2:",S2)

S2_length = len(S2)
print("Length of S2:",S2_length)

S1_new = S1[:N1]
S1_new_length = len(S1_new)
print("\nNew S1:",S1_new)
print("Length of new S1:",S1_new_length)

S2_new = S2[-N2:]
S2_new_length = len(S2_new)
print("\nNew S2:",S2_new)
print("Length of new S2:",S2_new_length)
print("\nNew string:",S1_new+" "+S2_new)

```

תרגיל מס' 25.

תפענח את הקוד הבא. תן מספר דוגמאות לפלט בהתאם לגנרטור של מספר אקראי.

```

import random
ru_letters =
u"абвгдежзийклмнопрстуфхцчшщъыьэюяАБВГДЕЖЗИЙКЛМНОПРСТУФХЦЧШЩЪЫЬЭЮЯ"
en_letters = u"abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ"
heb_letters = u"אבגדהוזחטיכלמנסעפצקרשת"
digits = u"0123456789"
letters = ru_letters + en_letters + heb_letters + digits
K = random.randrange(1,21)
S = ".join(random.choice(letters) for _ in range(K))
C = random.choice(S)
print("K:",K)
print("String:",S)
print("Character: ",C)
S_new = ""
for i in S:
    if i == C:
        S_new += i
    S_new += i
print("New string:",S_new)

```

תפענח את הקוד הבא. תן מספר דוגמאות לפלט בהתאם לגנרטור של מספר אקראי.

```
import random
ru_letters =
u"абвгдежзийклмнопрстуфхцчшщъыьэюяАБВГДЕЖЗИЙКЛМНОПРСТУФХЦЧШЩЪЫЬЭЮЯ"
en_letters = u"abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ"
heb_letters = u"אבגדהוזחטיכלמנסעפצקרשת"
digits = u"0123456789"
letters = ru_letters + en_letters + heb_letters + digits
K = random.randrange(1,21)
S = ".join(random.choice(letters) for _ in range(K))
C = random.choice(S)
S0 = '***'
print("K:",K)
print("String:",S)
print("S0:",S0)
print("Character: ",C)
S_new = ""
for i in S:
    if i == C:
        S_new += S0
    S_new += i
print("New string:",S_new)
```

תפענח את הקוד הבא. תן מספר דוגמאות לפלט בהתאם לגנרטור של מספר אקראי.

```
import random
ru_letters =
u"абвгдежзийклмнопрстуфхцчшщъыьэюяАБВГДЕЖЗИЙКЛМНОПРСТУФХЦЧШЩЪЫЬЭЮЯ"
en_letters = u"abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ"
heb_letters = u"אבגדהוזחטיכלמנסעפצקרשת"
digits = u"0123456789"
letters = ru_letters + en_letters + heb_letters + digits
K = random.randrange(1,21)
S = ".join(random.choice(letters) for _ in range(K))
C = random.choice(S)
S0 = '***'
print("K:",K)
print("String:",S)
print("S0:",S0)
print("Character: ",C)
```

```
S_new = ""
for i in S:
    S_new += i
    if i == C:
        S_new += S0
print("New string:", S_new)
```

תרגיל מס' 28.

תפענח את הקוד הבא. מה יהיה פלט של התוכנית?

```
str1 = 'Orcks'
print("Original String is", str1)
res = str1[0]
l = len(str1)
mi = int(l / 2)
res = res + str1[mi]
res = res + str1[l - 1]
print("New String:", res)
```

תרגיל מס' 29.

תפענח את הקוד הבא. מה יהיה פלט של התוכנית?

```
def get_ch(str1):
    print("Original String is", str1)
    mi = int(len(str1) / 2)
    res = str1[mi - 1:mi + 2]
    print("chars are:", res)
get_ch("WeWillRockYou")
get_ch("TheQueen")
```

תרגיל מס' 30.

תפענח את הקוד הבא. מה יהיה פלט של התוכנית?

```
def append_middle(s1, s2):
    print("Original Strings are", s1, s2)
    mi = int(len(s1) / 2)
    x = s1[:mi:]
    x = x + s2
    x = x + s1[mi:]
    print(" new string :", x)
append_middle("MySon", "Ido")
```



תרגיל מס' 31.

תפענח את הקוד הבא. מה יהיה פלט של התוכנית?

```
def mix_string(s1, s2):
    first_char = s1[0] + s2[0]
    middle_char = s1[int(len(s1) / 2):int(len(s1) / 2) + 1] + s2[int(len(s2) / 2):int(len(s2) / 2) + 1]
    last_char = s1[len(s1) - 1] + s2[len(s2) - 1]
    res = first_char + middle_char + last_char
    print("Mix String is ", res)

s1 = "America"
s2 = "Japan"
mix_string(s1, s2)
```

תרגיל מס' 32.

תפענח את הקוד הבא. מה יהיה פלט של התוכנית?

```
str1 = "WeAreTheChampionS"
print('Original String:', str1)
lower = []
upper = []
for char in str1:
    if char.islower():
        lower.append(char)
    else:
        upper.append(char)
sorted_str = ".join(lower + upper)
print('Result:', sorted_str)
```

תרגיל מס' 33.

תפענח את הקוד הבא. מה יהיה פלט של התוכנית?

```
def find_digits_chars_symbols(sample_str):
    char_count = 0
    digit_count = 0
    symbol_count = 0
    for char in sample_str:
        if char.isalpha():
            char_count += 1
        elif char.isdigit():
            digit_count += 1
        else:
```

```

        symbol_count += 1

    print("Chars =", char_count, "Digits =", digit_count, "Symbol =", symbol_count)

sample_str = "WWW.elecstudy.org_24022022!!!!"
print("total counts of chars, Digits, and symbols \n")
find_digits_chars_symbols(sample_str)

```

תרגיל מס' 34.

תפענח את הקוד הבא. מה יהיה פלט של התוכנית?

```

s1 = "AbcD"
s2 = "WxyZ"
s1_length = len(s1)
s2_length = len(s2)
length = s1_length if s1_length > s2_length else s2_length
result = ""
s2 = s2[::-1]
for i in range(length):
    if i < s1_length:
        result = result + s1[i]
    if i < s2_length:
        result = result + s2[i]
print(result)

```

תרגיל מס' 35.

תפענח את הקוד הבא. מה יהיה פלט של התוכנית?

```

def string_balance_test(s1, s2):
    flag = True
    for char in s1:
        if char in s2:
            continue
        else:
            flag = False
    return flag

s1 = "Love"
s2 = "AllYouNeedIsLove"
flag = string_balance_test(s1, s2)
print("s1 and s2 are balanced:", flag)

s1 = "Love!"
s2 = "AllYouNeedIsLove"
flag = string_balance_test(s1, s2)
print("s1 and s2 are balanced:", flag)

```

תרגיל מס' 36.

תפענח את הקוד הבא. מה יהיה פלט של התוכנית? כתוב תוכנית מחדש ללא שימוש בפקודות מיוחדות של מחרוזת.

```
str1 = "Isra Air is the Israel air flight company. IsraAir "  
sub_string = "Air"  
temp_str = str1.lower()  
count = temp_str.count(sub_string.lower())  
print("The count is:", count)
```

תרגיל מס' 37.

תפענח את הקוד הבא. מה יהיה פלט של התוכנית? כתוב תוכנית מחדש ללא שימוש בפקודות מיוחדות של מחרוזת.

```
input_str = "markalex012@gmail.com"  
total = 0  
cnt = 0  
for char in input_str:  
    if char.isdigit():  
        total += int(char)  
        cnt += 1  
avg = total / cnt  
print("Sum is:", total, "Average is ", avg)
```

תרגיל מס' 38.

תפענח את הקוד הבא. מה יהיה פלט של התוכנית? כתוב תוכנית מחדש ללא שימוש בפקודות מיוחדות של מחרוזת.

```
str1 = "markalex012@gmail.com"  
char_dict = dict()  
for char in str1:  
    count = str1.count(char)  
    char_dict[char] = count  
print('Result:', char_dict)
```

תרגיל מס' 39.

תפענח את הקוד הבא. מה יהיה פלט של התוכנית? כתוב תוכנית מחדש ללא שימוש בפקודות מיוחדות של מחרוזת.

```
str1 = "123456789"  
print("Original String 1 is:", str1)
```

```

str2 = "Electronics"
print("Original String is:", str1)

str1 = str1[::-1]
print("Final String is:", str1)
str2 = ".join(reversed(str2))
print("Final String is:", str2)

```

תרגיל מס' 40.

תפענח את הקוד הבא. מה יהיה פלט של התוכנית? כתוב תוכנית מחדש ללא שימוש בפקודות מיוחדות של מחרוזת.

```

str1 = "This is the simple program. Rewrite the program for the same aim. "
print("Original String is:", str1)
index = str1.rfind("the")
print("at index:", index)

```

תרגיל מס' 41.

תפענח את הקוד הבא. מה יהיה פלט של התוכנית? כתוב תוכנית מחדש ללא שימוש בפקודות מיוחדות של מחרוזת.

```

str1 = "This-is-the-simple-program."
print("Original String is:", str1)
sub_strings = str1.split("-")
print("Result:")
for sub in sub_strings:
    print(sub)
str2 = "Rewrite the program for the same aim"
print("Original String is:", str2)
sub_strings = str2.split(" ")
print("Result:")
for sub in sub_strings:
    print(sub)

```

תרגיל מס' 42.

תפענח את הקוד הבא. מה הוא פלט במקרה שקלט הוא: "abc83 cde7 1 b 24"?

```

s = input()
word_list = s.split()
num_list = []
for word in word_list:
    if word.isnumeric():
        num_list.append(int(word))
print(num_list)

```

תרגיל מס' 43.

תפענח את הקוד הבא. מה יהיה פלט של התוכנית? כתוב תוכנית מחדש ללא שימוש בפקודות מיוחדות של מחרוזת.

```
import string
str1 = "/*We study @Python & Micro_Processors"
print("Original string is ", str1)
new_str = str1.translate(str.maketrans(" ", "", string.punctuation))
print("New string is ", new_str)
```

תרגיל מס' 44.

תפענח את הקוד הבא. מה הוא פלט במקרה שקלט הוא: "0 It's 1Is 2Wonderful 3World. 4Yes"? ממש אותה תוכנית ללא פעולות מיוחדות של מחרוזות.

```
string = input()
length = len(string)
lower = upper = 0
for i in string:
    if i.islower():
        lower += 1
    elif i.isupper():
        upper += 1
per_lower = lower / length * 100
per_upper = upper / length * 100
per_other = (length - lower - upper) / length * 100
print("Lower: %.2f%%" % per_lower)
print("Upper: %.2f%%" % per_upper)
print("Other : %.2f%%" % per_other)
```

תרגיל מס' 45.

תפענח את הקוד הבא. מה יהיה פלט של התוכנית? כתוב תוכנית מחדש ללא שימוש בפקודות מיוחדות של מחרוזת.

```
str1 = "markalex012@gmail.com is my 3w.elecstudy"
print("The original string is : " + str1)
res = []
temp = str1.split()
for item in temp:
    if any(char.isalpha() for char in item) and any(char.isdigit() for char in item):
        res.append(item)
print("Displaying words with alphabets and numbers")
for i in res:
    print(i)
```

תרגיל מס' 46.

תפענח את הקוד הבא. תן מספר דוגמאות של פלט בהתאם לקלט שונה.

```
s = input()
l = len(s)
l = l//2
for i in range(l):
    if s[i] != s[-1-i]:
        print("No")
        quit()
print("Yes")
```

תרגיל מס' 47.

תפענח את הקוד הבא. תן מספר דוגמאות של פלט בהתאם לקלט שונה.

```
extensions = ['.png', '.jpg', '.jpeg', '.gif', '.svg']
file = input().split('.')
if len(file) >= 2:
    fileExtension = file[-1].lower()
    if fileExtension in extensions:
        print("Yes")
    else:
        print("No")
else:
    print("The file doesn't have an extension")
```

תרגיל מס' 48.

תפענח את הקוד הבא. מה הוא פלט במקרה שקלט הוא: " abc83 cde7 1 b 24 " ?

```
a = input()
num_list = []
num = ""
for char in a:
    if char.isdigit():
        num = num + char
    else:
        if num != "":
            num_list.append(int(num))
            num = ""
if num != "":
    num_list.append(int(num))
print(num_list)
```

תרגיל מס' 49.

תפענח את הקוד הבא. מה הוא פלט במקרה שקלט הוא: "0 It's 1A 2Wonderful 3World. 4Yes "?  
ממש אותה תוכנית עם פעולות מיוחדות של מחרוזות.

```
string = input("Enter the text: ")
length = len(string)
let_s = let_b = 0
for i in string:
    if 'a' <= i <= 'z':
        let_s += 1
    elif 'A' <= i <= 'Z':
        let_b += 1
print("%% Lower Case: %.2f" % (let_s / length * 100))
print("%% Upper case: %.2f" % (let_b / length * 100))
print("%% Other symbols: %.2f" % ((length-let_s-let_b)/length*100))
```

תרגיל מס' 50.

תפענח את הקוד הבא. מה יהיה פלט של התוכנית? כתוב תוכנית מחדש ללא שימוש בפקודות מיוחדות של מחרוזות.

```
str1 = 'The first x86 processor was 8086 processor'
print("Original string is", str1)
res = "".join([item for item in str1 if item.isdigit()])
print(res)
```

תרגיל מס' 51.

תפענח את הקוד הבא. מה הוא פלט במקרה שקלט הוא: " It's a wonderful world. Yes "?  
כתוב תוכנית מחדש ללא שימוש בפקודות מיוחדות של מחרוזות.

```
s = input()
l = s.split()
s1 = '*'.join(l)
print(s1)
```

תרגיל מס' 52.

תפענח את הקוד הבא. מה הוא פלט במקרה שקלט הוא: " It's a wonderful world. Yes "?  
תוכנית מחדש ללא שימוש בפקודות מיוחדות של מחרוזות.

```
string = input()
words = string.split()
shortest = min([len(word) for word in words])
print(shortest)
```

תרגיל מס' 53.

תפענח את הקוד הבא. מה הוא פלט במקרה שקלט הוא: " It's a wonderful world. Yes ?" כתוב תוכנית מחדש ללא שימוש בפקודות מיוחדות של מחרוזת.

```
s = input()
word_list = s.split()
print(word_list)
word_list.reverse()
print(word_list)
s2 = " ".join(word_list)
print(s2)
```

תרגיל מס' 54.

תפענח את הקוד הבא. מה עושה תוכנית? כתוב תוכנית מחדש ללא שימוש בפקודות מיוחדות של מחרוזת.

```
things = "tree, box, chair, lamp, \n" \
"desk, cat, dog, grass, \n" \
"pig, box, lamp, shelf"
print(things)
print()
old_item = input("Old item: ")
new_item = input("New item: ")
len_old_item = len(old_item)
i = things.find(old_item)
while i > 0:
    before = things[:i]
    after = things[i+len_old_item:]
    things = before + new_item + after
    i = things.find(old_item)
print()
print(things)
```

תרגיל מס' 55.

תפענח את הקוד הבא. מה הוא פלט במקרה שקלט הוא: " It's a wonderful world. Yes ?" כתוב תוכנית מחדש עם שימוש בפקודות מיוחדות של מחרוזת.

```
s = input()
i = 0
while s[i] == ' ':
    i += 1
s = s[i:]
```



```

i = len(s)
while s[i - 1] == ' ':
    i -= 1
s = s[:i]
s_new = s[0]
i = 1
while i < len(s):
    if s[i] != ' ':
        s_new += s[i]
    elif s[i - 1] != ' ':
        s_new += '*'
    i += 1
print(s_new + '!')

```

תרגיל מס' 56.

תפענח את הקוד הבא. מה הוא פלט במקרה שקלט הוא: " It's a wonderful world. Yes " ? כתוב תוכנית מחדש ללא שימוש בפקודות מיוחדות של מחרוזת.

```

string = input()
words = string.split()
shortest = words[0]
for i in words[1:]:
    if len(i) < len(shortest):
        shortest = i
print(shortest)
print(len(shortest))

```

תרגיל מס' 57.

תפענח את הקוד הבא. מה הוא פלט במקרה שקלט הוא: " abc83 cde7 1 b 24 " ?

```

s = input()
l = len(s)
integ = []
i = 0
while i < l:
    s_int = ""
    a = s[i]
    while '0' <= a <= '9':
        s_int += a
    i += 1
    if i < l:
        a = s[i]
    else:

```

```

break
i += 1
if s_int != "":
    integ.append(int(s_int))
print(integ)

```

תרגיל מס' 58.

תפענח את הקוד הבא. מה הוא פלט במקרה שקלט הוא: " It's a wonderful world. Yes ?" ממש אותה תוכנית ללא פעולות מיוחדות של מחרוזות.

```

s = input()
s = s.split()
s.sort(key=len)
s = " ".join(s)
print(s)

```

תרגיל מס' 59.

תפענח את הקוד הבא. מה יהיה פלט של התוכנית?

```

str_list = ["Israel", "USA", "", "England", None, "Ukraine", "", "Poland"]
res_list = []
for s in str_list:
    if s:
        res_list.append(s)
print(res_list)

```

תרגיל מס' 60.

תפענח את הקוד הבא. מה הוא פלט במקרה שקלט הוא: "It too much horror".

```

s = input()
new_s = ""
for i in s:
    if i not in new_s and i != ' ':
        new_s += i
print(new_s)

```

תרגול בנושא כתיבת קוד.

**נא לפתור כל תרגיל ב2 דרכים – בעזרת פקודות מיוחדות של מחרוזת וללא פקודות מיוחדות.**

תרגיל מס' 1.

כתוב תוכנית הקולטת מחרוזת מהמשתמש ומדפיסה תו שני ותו לפני אחרון של המחרוזת.

תרגיל מס' 2.

כתוב תוכנית הקולטת מחרוזת מהמשתמש ומציגה אותה בסדר הפוך.

תרגיל מס' 3.

כתוב תוכנית הקולטת מחרוזת מהמשתמש ומדפיסה אותה כך שאחרי כל תו מופיע רווח.

תרגיל מס' 4.

כתוב תוכנית הקולטת מחרוזת מהמשתמש ומדפיסה אותה כך שאחרי כל אות מופיע רווח ואחרי כל ספרה סימן קריאה ואחרי כל תו אחר סימן שאלה.

תרגיל מס' 5.

כתוב תוכנית הקולטת מחרוזת מהמשתמש ומחליפה את המילה הארוכה ביותר בסימני \* לפי כמות אותיות שיש במילה הזאת.

תרגיל מס' 6.

כתוב תוכנית הקולטת מחרוזת מהמשתמש ומחליפה את המילה הקצרה ביותר (אבל לא מחרוזת ריקה) בסימני \* לפי כמות אותיות שיש במילה הזאת.

תרגיל מס' 7.

כתוב תוכנית הקולטת מחרוזת מהמשתמש ומחליפה את המילה הארוכה ביותר בסימני \* לפי כמות אותיות שיש במילה הזאת ואת המילה הקצרה ביותר (אבל לא מחרוזת ריקה) בסימני + לפי כמות אותיות שיש במילה הזאת.

תרגיל מס' 8.

כתוב תוכנית הקולטת מחרוזת מהמשתמש וסופרת כמה ספרות וכמה אותיות (קטנות וגדולות) יש במחרוזת.

תרגיל מס' 9.

כתוב תוכנית הקולטת מחרוזת מהמשתמש וסופרת כמה אותיות קטנות וכמה אותיות גדולות יש במחרוזת.

תרגיל מס' 10.

כתוב תוכנית הקולטת מחרוזת מהמשתמש וסופרת כמה ספרות, כמה סימנים וכמה אותיות קטנות וכמה אותיות גדולות יש במחרוזת.

תרגיל מס' 11.

כתוב תוכנית הקולטת מחרוזת מהמשתמש ומעתיקה את כל האותיות למחרוזת בשם letters, את כל הספרות למחרוזת nums, ואת כל הסימנים למחרוזת signs. בסוף התוכנית היא מדפיסה כל מחרוזת בשורה חדשה עם מספר שמייצג כמות תווים בכל מחרוזת.

תרגיל מס' 12.

כתוב תוכנית הקולטת מחרוזת מהמשתמש והופכת אותיות קטנות לאותיות גדולות ולהפך. בסוף התהליך תוכנה מדפיסה גם מחרוזת המקור וגם מחרוזת היעד.

תרגיל מס' 13.

כתוב תוכנית הקולטת מחרוזת מהמשתמש. תוכנית בודקת האם אפשר להפוך את המחרוזת למספר שלם, מספר ממשי או למספר מרוכב. אם כן מדפיסה את המספר ואת סוג שלו. (לדוגמה: "1233" מוציא "int 1233", "-1.233" מוציא "float -1.223", "2+3j" או "2+3i" מוציא "complex 2+3i").

תרגיל מס' 14.

כתוב תוכנית הקולטת מהמשתמש מספר כלשהו והופכת אותו למחרוזת תווים ומדפיסה אותו.

תרגיל מס' 15.

כתוב תוכנית הקולטת מהמשתמש מספר כלשהו והופכת אותו קודם כל לבסיס 2 ואז למחרוזת תווים ומדפיסה אותו.

תרגיל מס' 16.

כתוב תוכנית הקולטת מהמשתמש מספר כלשהו והופכת אותו קודם כל לבסיס 16 ואז למחרוזת תווים ומדפיסה אותו.

תרגיל מס' 17.

כתוב תוכנית הקולטת מהמשתמש מספר כלשהו והופכת אותו למחרוזת תווים ומדפיסה אותו בסדר ספרות הפוך.

תרגיל מס' 18.

כתוב תוכנית הקולטת מהמשתמש מספר כלשהו, מחשבת סכום ספרות ומדפיסה את סכום בצורה מילולית.

תרגיל מס' 19.

כתוב תוכנית הקולטת מהמשתמש מחרוזת בתור מספר בבסיס 2 והופכת אותו למספר בבסיס 10 ואז מדפיסה אותו.

תרגיל מס' 20.

כתוב תוכנית הקולטת מהמשתמש מחרוזת בתור מספר בבסיס 16 והופכת אותו למספר בבסיס 10 ואז מדפיסה אותו. (שים לב שמחרוזת בבסיס 16 יכולה להכיל גם אותיות גדולות וגם קטנות).

תרגיל מס' 21.

כתוב תוכנית הקולטת מחרוזת מהמשתמש. יש לבדוק האם המחרוזת היא "פלינדרום", זאת אומרת האם ניתן לקרוא מחרוזת זו מימין לשמאל או משמאל לימין באופן זהה. לדוגמה: "שמש", "תודות", "12321" וכו'.

תרגיל מס' 22.

כתוב תוכנית הקולטת מחרוזת מהמשתמש ומספר N כלשהו. אם מחרוזת גדולה ממספר N יש לצמצם אותה, כך שהיא תהיה בדיוק לפי אורך של N (תצמצם תווים בסוף המחרוזת), אם מחרוזת קצרה יותר ממספר N, תוסיף רווחים בסוף המחרוזת.

תרגיל מס' 23.

כתוב תוכנית הקולטת מחרוזת מהמשתמש ובודקת איזה תווים וכמה תווים מכל סוג יש. לדוגמה: מחרוזת של משתמש: "Let it be!"  
נקבל:

l - 1, e - 2, t - 2, ' ' - 2, i - 1, b - 1, ! - 1

תרגיל מס' 24.

כתוב תוכנית הקולטת מהמשתמש 2 מחרוזות ובודקת האם מחרוזות שוות או לא.

תרגיל מס' 25.

כתוב תוכנית הקולטת מהמשתמש 2 מחרוזות ובודקת האם מחרוזות שוות או לא. אם לא תציג תווים שזהים ב-2 מחרוזות באותם מקומות.

תרגיל מס' 26.

כתוב תוכנית הקולטת מהמשתמש 2 מחרוזות ובודקת האם מחרוזות שוות או לא. אם לא תציג תווים שזהים ב-2 מחרוזות אפילו שהם עומדים במקומות שונים.

תרגיל מס' 27.

כתוב תוכנית הקולטת מהמשתמש שם של קובץ וסיומת הקובץ. יש לבדוק האם הסיומת היא חוקית: txt, jpg, bmp, ....

תרגיל מס' 28.

כתוב תוכנית הקולטת מחרוזת מהמשתמש. תוכנית צריכה לסדר אותה כך שמילים ארוכות יותר יהיו בתחילת המחרוזת ומילים קצרות יותר יהיו בסוף המחרוזת.

כדי לפתור בעיה זו, קל יותר להשתמש בפונקציות המחרוזת המובנות של שפת פייתון ובשיטות טיפול ברשימות.

1. באמצעות שיטת `split()`, נחלק את המחרוזת למילים לפי רווחים.

2. נשתמש בפונקציה `sorted()` או `sort()` כדי למיין את אורך המחרוזת. הפונקציה `len()` מחשבת אורך המילה. ההבדל בין `sorted()` לבין `sort()` הוא שהראשונה מחזירה רשימה חדשה, והשנייה ממיינת במקום, כלומר משנה את הרשימה הקיימת.

3. בשיטת מחרוזת join() צרו מחרוזת חדשה מהרשימה.

תרגיל מס' 29.

כתוב תוכנית הקולטת מחרוזת מהמשתמש. תוכנית צריכה לבדוק התפלגות באחוזים של אותיות גדולות לעומת אותיות קטנות.

כידוע בקידודים, התווים מסודרים, וקוד התו 'a' פחות מהדמות 'b'. לכן, אם התו הבא שייך לטווח 'a' ל- 'z', הרי שמדובר באות קטנה; אם הטווח הוא בין 'A' ל- 'Z', הרי שהוא גדול.

תרגיל מס' 30.

כתוב תוכנית הקולטת מהמשתמש מחרוזת ואות. צריכים למצוא את המילה (מילים) שמכילה (מכילים) אות הזאת.

בפייתון, סוג נתוני מחרוזת כולל שיטת split() המפרידה בין מילים לפי רווח או תו אחר, אם מועברים כארגומנט. נקבל רשימת מילים.

בהמשך מרשימה זו תוכלו לחלץ מילים בודדות על ידי גישה אליהם לפי אינדקס. התוכנית שלהלן מניחה תחילה שהמילה הראשונה ברשימה היא הקצרה ביותר. המילים הנותרות משולבות בתעלות, כאשר אורכן משווה למילה במשתנה הקצר ביותר. אם המילה הנוכחית קצרה יותר, הערך הקצר ביותר מוחלף.

תרגיל מס' 31.

כתוב תוכנית הקולטת מהמשתמש מחרוזת ואות. תוכנית צריכה להדפיס אותה בסדר הפוך של המילים.

כדי לפתור בעיה זו, יש להשתמש בשיטות טיפול במחרוזות split () ו-join(), גם כן בשיטה reverse(). שיטת split () ממירה את המחרוזת המקורית לרשימת מחרוזות, reverse() הופכת את רשימת המילים ו-join() משלבת את המילים ברשימה שוב למחרוזת יחידה.

תרגיל מס' 32.

כתוב תוכנית הקולטת מהמשתמש מחרוזת, תת מחרוזת מקור ותת מחרוזת היעד. תוכנית צריכה למצוא תת מחרוזת מקור ולהחליף אותה בתת מחרוזת היעד. (אם יש מספר תת מחרוזות היעד יש להחליף את כולם).

למחרוזות יש שיטת חיפוש find() המחזירה את האינדקס של התו הראשון של התת מחרוזת במחרוזת, או -1 אם התת מחרוזת לא נמצאת.

האלגוריתם לפתרון הבעיה עשוי להיות כדלקמן:

1. מצא את פעם הראשונה שתת מחרוזת מופיעה במחרוזת.

2. אנו לוקחים פריסה מתחילת הקו לתחילת המחרוזת הישנה.

3. אנו לוקחים פריסה מקצה המחרוזת הישנה לקצה המחרוזת.

4. נצרף תחילת המחרוזת, מחרוזת חדשה, סוף המחרוזת.

5. שוב אנו בודקים במחרוזת את התרחשות התת מחרוזת הישנה בה.

תרגיל מס' 33.

כתוב תוכנית הקולטת מהמשתמש מחרוזת ואות. תוכנית צריכה להחליף את כל הרווחים בסימן "\*" .  
אם יש מספר רווחים ברצף, מחליפים אותם ב-"\*" בודד.

החלף רווחים במחרוזת ב- \* . אם ישנם מספר רווחים ברצף, יש להחליף אותם בסינגל "\*" , הסר את הרווחים בתחילת המחרוזת ובסוף.

פתרון של בעיה זו בצורה קלאסית, מבלי להשתמש בתכונות המתקדמות של פייתון, עשוי להיות כמתואר:

ראשית אנו נפטרים מהמרחבים בתחילת הקו ובסיומו, אם בכלל. לשם כך, חזרו על המחרוזת אופי אחר תו תחילה או מהסוף. ברגע שמתרחשת הדמות הראשונה שאינה מלבנית, אנו לוקחים ממנה פרוסה לקצה הקו או להתחלה, במקרה של הסרת רווחים מהסוף.

בשלב הבא אנו שוב ממיינים את תו המחרוזת לפי תו. אם התו הבא אינו רווח, צרו שורה חדשה על ידי הוספת תו זה אליו. תווי רווח נופלים לענף elif, אך הוא בודק אם הסימן הקודם היה שווה לרווח. אם זה לא המקרה (זה לא היה שווה), רק "\*" מתווסף לשורה החדשה. דילוג על רווחים נוספים יהיה פשוט ולא יתווסף למחרוזת החדשה.

פתרון קלסי של שפת פייתון:

בשפת התכנות Python, קל יותר לפתור משימה זו על ידי המרת מחרוזת לרשימה. שיטת split() מפצלת מחרוזת למילים לפי רווחים. לא משנה כמה מהם. ואז נותר רק לאסוף את המילים במחרוזת שוב. זה יותר קל לעשות זאת בשיטת מחרוזת join() .

תרגיל מס' 34.

כתוב תוכנית הקולטת מהמשתמש מחרוזת שמורכבת מאותיות וספרות. תוכנית צריכה לבנות רשימה שמורכבת מכל המספרים שנמצאים במחרוזת המקור. לדוגמה: "abc83 cde7 1 b 24" -> [83, 7, 1, 24].

לבחור מהמחרוזת מספרים.

ניתנה מחרוזת הכוללת מספרים טבעיים ומילים. יש צורך ליצור רשימת מספרים הכלולים בשורה זו. לדוגמה, המיתר "abc83 cde7 1 b 24" ניתן. בפלט עלינו לקבל רשימה [83, 7, 1, 24].

נמייך את השורה תו אחרי תו. אם התו הבא הוא ספרה, יש להוסיף אותה לשורה חדשה. בשלב הבא, בדוק את התווים שמאחוריו, ואם הם גם מספרים, הוסיף אותם לסוף מחרוזת המספרים החדשה הזו. כאשר התו הבא אינו ספרה, או שמגיעים לסוף השורה, יש להמיר את המחרוזת ממספרים למספרים ולהוסיף לרשימה.

תרגיל מס' 34.

קולטים מחרוזת. אנו צריכים להוריד ממנה את כל הרווחים ואז להוריד את כל התווים זהים ברצף. לדוגמה: "abc cde def" -> "abcdef".

הדרך הקלה ביותר לפתור את הבעיה היא אם יוצרים את המחרוזת המתקבלת במשתנה אחר, ולא תשנה את ערך המשתנה שאליו הוקצה המחרוזת שהזנה.

אחזר כל תו של המחרוזת שהזנה. אם הוא אינו מופיע בשורה חדשה ואינו רווח, יש להוסיף אותו לסוף הקו החדש.

תרגיל מס' 35.

קולטים מחרוזת. אנו צריכים להוריד ממנה את כל הרווחים ואז להוריד את כל התווים זהים ברצף ללא מחרוזת נוספת (בתוך אותה מחרוזת). לדוגמה: "abc cde def" -> "abcdef".

תרגיל מס' 36.

כתוב תוכנית הקולטת מהמשתמש מחרוזת ומילה (תת מחרוזת). תוכנית צריכה לבדוק האם מילה קיימת בתוך מחרוזת. אם כן, כמה פעמים.

לדוגמה:

מחרוזת:

And all he said--that it was necessary to await provisions, or that the men had no boots--was so simple, while what they proposed was so complicated and clever, that it was evident that he was old and stupid and that they, though not in power, were commanders of genius.

מילה:

it

לאחר בדיקה נראה שמילה מופיעה 2 (לא נחשב מילה await למרות שבתוך מילה הזאת גם מופיע מילה it).

And all he said--that it was necessary to await provisions, or that the men had no boots--was so simple, while what they proposed was so complicated and clever, that it was evident that he was old and stupid and that they, though not in power, were commanders of genius.

תרגיל מס' 37.

כתוב תוכנית שלוקחת טקסט מוגדר מראש:

All that was done around her and to her at this time, all the attention devoted to her by so many clever men and expressed in such pleasant, refined ways, and the state of dove-like purity she was now in (she wore only white dresses and white ribbons all that time) gave her pleasure, but her pleasure did not cause her for a moment to forget her aim.

ומדפיסה התפלגות אותיות מגדולה ביותר לקטנה ביותר. תוכנית סופרת כמה פעמים מופיע כל אות (לא משנה קטנה או גדולה) ואז מסדרת כמויות בסדר יורד.

תרגיל מס' 38.

כתוב תוכנית שלוקחת טקסט מוגדר מראש:

His second wife, Madame de Montesson, whom he married secretly in 1773, was a clever woman and an authoress of some repute.

ומסדרת אותו מחדש לפי "צופן של קיסר יוליוס" ומדפיסה אותו מחדש.

His second wife,	H	c	i	d	e	e	.		
Madame de Montesson,	i	o	f	a		s	.		



whom he married secretly	s	n	e	m	M	s	.		
in 1773, was a clever		d	,	e	o	o	.		
woman and an authoress	s		M		n	n			
of some repute.	e	w	a	d	t	.			

תרגיל מס' 39.

כתוב תוכנית שלוקחת טקסט מוגדר מראש:

We use both first and third-party cookies to personalise web content. we analyse visits to our websites and tailor advertisements. Some of these cookies are necessary for the website to function! whilst others require your consent. Are you shure? more detail can be found in our cookie policy and you can tailor your choices in the preference centre.

תוכנית צריכה לבדוק שמשפט מתחיל באות גדולה. (משפט ראשון בתחילת טקסט, כל היתר משפטים מתחילים אחרי רווח שבא לאחר אחד מהסימנים הבאים: ". ! ?".

## פרק י'. מבנה נתונים מסוג Tuple - צירוף.

### מה זה Tuple - צירוף

צירוף – זה סוג נתונים כמו רשימות עם הבדל יחיד – זה מבנה נתונים שלא ניתן לשינוי לאחר הגדרה הראשונה. כמו שראינו רשימות מגדירים עם סוגריים "[]" ו צירוף מגדירים עם סוגריים "()". ניתן להחליף את כל הצירוף.

```
>>> a=(10,2.13,"square",-15,'c')
>>> a
(10, 2.13, 'square', -15, 'c')
>>> a[3]
-15
>>> a[1:3]
(2.13, 'square')
>>> a[0]=1
Traceback (most recent call last):
  File "<pyshell#4>", line 1, in <module>
    a[0]=1
TypeError: 'tuple' object does not support item assignment
>>> a=(10,2,3,4,5,6)
>>> a
(10, 2, 3, 4, 5, 6)
```

לצירוף יש 4 תכונות:

- סדר

צירוף, זה סוג נתונים, שכל הנתונים מוכנסים לפי סדר ולכל אחד יש מיקום ותוכן.

- לא ניתן לשינוי

לאחר הגדרת צירוף אין אפשרות להוסיף או להוריד איברים מתוך הצירוף.

- הטרוגניות של נתונים

צירוף יכול להכיל נתונים מסוגים שונים.

- כפילויות נתונים

ניתן להכניס ערכים שחוזרים על עצמם, ואפילו מספר מעמים.

### הגדרת הצירוף.

ישנם 2 שיטות להגדרת צירוף: בשיטה ראשונה פשוט נרשום איברי הצירוף בסוגריים: a=(8,9) ובשיטה שנייה נשתמש במילה שמורה d=tuple((30, 45.75)).

```
number_tuple = (10, 20, 25.75)
print(number_tuple)
string_tuple = ('Jessa', 'Emma', 'Kelly')
print(string_tuple)
sample_tuple = ('Jessa', 30, 45.75, [25, 78])
```

```
print(sample_tuple)
sample_tuple2 = tuple(('Jessa', 30, 45.75, [23, 78]))
print(sample_tuple2)
```

עולה שאלה סבירה. מדוע סוג הנתונים הזה הוצג בשפת התכנות, שהיא בעצם רשימה בלתי ניתנת לשינוי? הסיבה היא שלפעמים צריך להגן על הרשימה מפני שינויים. המרה של הצירוף לרשימה, במידת הצורך, כמו גם בצע את פעולת ההפוך בקלות בעזרת פונקציות list() ו-tuple() המובנות בפייתון:

```
>>> a=(8,9)
>>> b=[1,2,3,4]
>>> c=list(a)
>>> d=tuple(b)
>>> c
[8, 9]
>>> d
(1, 2, 3, 4)
```

## גישה לאיברים של צירוף.

אינדקסים של איברים של צירוף הם כמו אינדקסים של רשימות או איברים של מערך. ניתן לגשת לכל אינדקס גם מהתחלה וגם מהסוף. שימו לב שמהתחלה אינדקס ראשון הוא 0, ומהסוף אינדקס ראשון הוא -1.

	P	Y	T	H	O	N	
Positive Indexing →	0	1	2	3	4	5	
	-6	-5	-4	-3	-2	-1	← Negative Indexing

לא ניתן לשנות איברים של צירוף, אבל ניתן לגשת לצירוף איבר-איבר כדי לבדוק מה תוכן שלו.

```
sample_tuple = tuple((1, 2, 3, "Hello", [4, 8, 16]))
for item in sample_tuple:
    print(item)
for item in sample_tuple:
    sample_tuple[item]+=1
```

```
1
2
3
```

```

Hello
[4, 8, 16]
Traceback (most recent call last):
  File "C:\Users\marka\PycharmProjects\Game\main.py", line 5, in <module>
    sample_tuple[item]+=1
TypeError: 'tuple' object does not support item assignment

```

ישנה אופציה לגשת לאיברים בודדים או "פרוסות".

קוד	הסבר
<pre> tuple1=('P', 'Y', 'T', 'H', 'O', 'N') for i in range(4):     print(tuple1[i])  print(tuple1[7])  print(tuple1[2.0])  print(tuple1[-1]) print(tuple1[-2])  for i in range(-6, 0):     print(tuple1[i], end=" ") </pre>	<p>הגדרת צירוף נראה על המסך:</p> <p>P Y T H</p> <p>הודעת שגיאה שמתון נמצא מעבר לאורך של צירוף.          IndexError: tuple index out of range</p> <p>הודעת שגיאה שמיקום באיבר יכול להיות רק שלם.          tuple indices must be integers or slices, not float</p> <p>גישה לאיבר אחרון 'N'          גישה לאיבר לפני אחרון 'O'</p> <p>הדפסת איברים במיקום -6 ועד למיקום 0 (לא כולל)          P, Y, T, H, O, N,</p>

נבדוק עוד מספר דוגמאות של עבודה עם "פרוסות".

קוד	הסבר
<pre> tuple1 = (0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10) print(tuple1[1:5]) print(tuple1[:5]) print(tuple1[6:]) print(tuple1[-5:-1]) </pre>	<p>הגדרת צירוף</p> <p>(1, 2, 3, 4)          (0, 1, 2, 3, 4)          (6, 7, 8, 9, 10)          (6, 7, 8, 9)</p>

קחו בחשבון את המקרה כאשר ראוי להשתמש בצירוף. בפייתון, אובייקטים הניתנים להחלפה מועברים לפונקציה על ידי הפניה. משמעות הדבר היא שלא נוצר עותק של האובייקט, ולפרמטר מוקצה הפניה לאובייקט קיים. כתוצאה מכך, אם האובייקט משתנה בגוף הפונקציה, שינויים אלה נוגעים לאובייקט הגלובלי.

```
def addNum(seq, num):
    for i in range(len(seq)):
        seq[i] += num
    return seq

origin = [3, 6, 2, 6]
changed = addNum(origin, 3)
print(origin)
print(changed)
```

```
[6, 9, 5, 9]
[6, 9, 5, 9]
```

תוכנית זו שגויה. למרות שלא תתרחש שום זריקת חריגה, היא מכילה שגיאה לוגית. כלומר, גם הרשימה המקורית שונתה. פרמטר ה- seq לא הכיל קישור לרשימה המקומית שלו, אלא לרשימה המקורית. לפיכך, אין משמעות בהצגת ההחזרה. אם הפונקציה נועדה לשנות את הרשימה הגלובלית, התוכנית צריכה להיראות כך:

```
def addNum(seq, num):
    for i in range(len(seq)):
        seq[i] += num

origin = [3, 6, 2, 6]
addNum(origin, 3)
print(origin)
```

```
[6, 9, 5, 9]
```

מה לעשות אם עדיין לא צריך לשנות את הרשימה המקורית, אלא ליצור עליה חדשה? ניתן לפתור את הבעיה בכמה דרכים. ראשית, בפונקציה תוכלו ליצור ברשימה המקומית, ואז להחזיר אותה:

```
def addNum(seq, num):
    new_seq = []
    for i in seq:
        new_seq.append(i + num)
    return new_seq

origin = [3, 6, 2, 6]
changed = addNum(origin, 3)
print(origin)
print(changed)
```

```
[3, 6, 2, 6]
[6, 9, 5, 9]
```

הרשימה המקורית בפונקציה לא משתנה. האלמנטים שלה רק ממוינים. הדרך השנייה להגן על הרשימה המקורית היא להשתמש בצירוף. שיטה זו אמינה יותר, מכיוון שבתוכנות גדולות קשה לעקוב ולוודא שאף פונקציה לא מכילה פקודות לשינוי נתונים גלובליים. למרות שניתן להמיר לצביעה גם כשאתה עובר לפונקציה וגם בפונקציה עצמה, עדיף להפוך מיד את הרשימה הגלובלית לטפל. מכיוון שאובייקטים בלתי ניתנים להחלפה מועברים לפי ערך, לא על ידי הפניה, העתק של המבנה ולא המקורי ייכנס לפונקציה. גם אם המקור מועבר לשם, אי אפשר לשנות אותו. באפשרותך רק להעתיק אותו ו / או לשנות את הסוג, ובכך ליצור מבנה מקומי ולעשות איתו מה שבא לכם.

```
def addNum(seq, num):
    seq = list(seq)
    for i in range(len(seq)):
        seq[i] += num
    return seq

origin = (3, 6, 2, 6)
changed = addNum(origin, 3)
print(origin)
print(changed)
```

```
(3, 6, 2, 6)
[6, 9, 5, 9]
```

## שינוי והוספת איברים לצירוף.

כדי לשנות או להוסיף איברים לצירוף, אפשר להעביר אותו לרשימה, לשנות אותה ולהחזיר לצירוף.

קוד	הסבר
<pre>tuple1 = (0, 1, 2, 3, 4, 5) tuple1[1] = 10 sample_list = list(tuple1) sample_list.append(6) sample_list[3]=100 tuple1 = tuple(sample_list) print(tuple1)</pre>	הגדרת צירוף אי-אפשר לעשות את זה – נקבל הודעת שגיאה העברת צירוף לתוך רשימה הוספת נתון לרשימה שינוי נתון בתוך הרשימה העברת הרשימה בחזרה לצירוף הדפסת צירוף חדש

כדי להעתיק מצירוף לצירוף אפשר להשתמש באופרטור "=".

קוד	הסבר
-----	------

```

tuple1 = (0, 1, 2, 3, 4, 5)
tuple2 = tuple1
print(tuple2)
sample_list = list(tuple2)
sample_list.append(6)
tuple2 = tuple(sample_list)
print(tuple1)
print(tuple2)

```

הגדרת צירוף  
העתקה מצירוף לצירוף  
הדפסת צירוף שני  
העברת צירוף לתוך רשימה  
הוספת נתון לרשימה  
העברת הרשימה בחזרה לצירוף  
הדפסת הצירוף המקורי והצירוף היעד

3 שיטות ל שרשור צירופים.

שרשור צירופים בעזרת אופרטור "+", פונקציה sum ופונקציה chain מספריה itertools.

```

import itertools
tuple1 = (1, 2, 3, 4, 5)
tuple2 = (3, 4, 5, 6, 7)
tuple3 = tuple1 + tuple2
print(tuple3)
tuple3 = sum((tuple1, tuple2), ())
print(tuple3)
tuple3 = tuple(item for item in itertools.chain(tuple1, tuple2))
print(tuple3)

```

הגדרה והדפסה של צירוף שמכיל צירופים.

```

nested_tuple = ((20, 40, 60), (10, 30, 50), "Python")
print(nested_tuple[2][0])
for i in nested_tuple:
    print("tuple", i, "elements")
    for j in i:
        print(j, end=" ")
    print("\n")

```

```

C:\Users\marka\PycharmProjects\Game\venv\Scripts\python.exe
C:/Users/marka/PycharmProjects/Game/main.py
P
tuple (20, 40, 60) elements
20, 40, 60,
tuple (10, 30, 50) elements
10, 30, 50,
tuple Python elements
P, y, t, h, o, n,

```

## ניתן להגדיר רשימות בתוך צירוף.

במקרה כזה אפשר לשנות רשימה בתוך צירוף ולא את הצירוף עצמו.

```
>>> nested=(1,"do",["param",10,20])
>>> nested
(1, 'do', ['param', 10, 20])
>>> nested[2][1]=15
>>> nested
(1, 'do', ['param', 15, 20])
```

ביטויים בסגנון: `nested[2][1]` משמשים לגישה לאובייקטים מקוננים. האינדקס הראשון מציין את מיקום האובייקט המקונן, השני מציין את אינדקס האלמנט בתוך האובייקט המקונן. אז במקרה זה, ברשימה עצמה בתוך הכותרת יש אינדקס 2, ולפריט הרשימה 10 יש אינדקס 1 ברשימה. מצב מוזר. הצירוף אינו ניתן לשינוי, אך כולנו יכולים לשנות זאת. למעשה, הזר נשאר ללא שינוי. הוא פשוט לא מכיל את הרשימה עצמה, אלא קישור אליה. לא ניתן לשנות את זה. אבל ניתן לשנות את הרשימה עצמה. כדי להקל על ההבנה, אנו כותבים צירוף כך:

```
>>> qq="qwerty",10,20
>>> t=(1,"do",qq)
>>> t
(1, 'do', ['qwerty', 10, 20])
```

צירוף מכיל הפניה משתנה. לא ניתן לשנות אותו לקישור אחר אך צירוף אינו מכיל את הרשימה עצמה לכן ניתן לשנות אותו כרצונך:

```
>>> qq.pop(0)
'qwerty'
>>> qq
[10, 20]
>>> t
(1, 'do', [10, 20])
```

מצד שני צירוף לא ניתן לשינוי – אם נשנה משתנה בתוך צירוף הצירוף עדיין יישאר קבוע.

```
>>> qq
[10, 20]
>>> a = "Kat"
>>> t = (a, qq)
>>> t
('Kat', [10, 20])
>>> a="Hero"
```



```
>>> t
('Kat', [10, 20])
```

שימו לב שנתונים מועברים לפי הערך ולא כמו בשפת C לפי מצביע.

## מונחים "אריזה" ו"פריקה". (Packing and Unpacking).

קוד	הסבר
<pre>tuple1 = 1, 2, "Hello" print(tuple1)</pre>	<p>"אריזת" נתונים בודדים לצירוף פלט:</p> <pre>(1, 2, 'Hello')</pre>
<pre>print(type(tuple1))</pre>	<p>אם נדפיס סוג של נתון זה נראה:</p> <pre>&lt;class 'tuple'&gt;</pre>
<pre>i, j, k = tuple1 print(i, j, k)</pre>	<p>כדי לפתוח או "לפרוק" צירוף ניתן משתנים לכל ערך שנמצא בתוך צירוף:</p> <pre>1 2 Hello</pre>
<pre>i, j = tuple1 print(i, j)</pre>	<p>גם במקרה של מעט נתונים וגם במקרה של כמות נתונים מעבר ממה שיש בצירוף, נקבל הודעת שגיאה.</p>
<pre>i, j, k, l = tuple1 print(i, j, k, l)</pre>	<p>Traceback (most recent call last): File "C:\Users\marka\PycharmProjects\Game\main.py", line 6, in &lt;module&gt; i, j = tuple1 ValueError: too many values to unpack (expected 2)</p> <p>Traceback (most recent call last): File "C:\Users\marka\PycharmProjects\Game\main.py", line 6, in &lt;module&gt; i, j, k, l = tuple1 ValueError: not enough values to unpack (expected 4, got 3)</p>

## כדי להימנע משינוי רשימות המקוריות, לא חייבים להשתמש בצירוף.

ניתן ליצור עותק בשיטת רשימת `copy()`, או לקחת פרוסה מההתחלה ועד הסוף. שינוי העותקים אינו משפיע על המקור בשום צורה.

דוגמה של מילוי צירוף בעזרת משפט `.for`.

```
>>> c=[(i+1, (i+1)/2) for i in range (4)]
>>> c
[(1, 0.5), (2, 1.0), (3, 1.5), (4, 2.0)]
```

דוגמה עם `for` כפול ו-`.if`.

```
>>> a=[1,2,3]
>>> b=[4,5,6]
>>> [(i,j) for i in a for j in b if i*j<=10]
[(1, 4), (1, 5), (1, 6), (2, 4), (2, 5)]
```

חיבור של אלמנטים של צירוף והעתקה שלהם לתוך רשימה חדשה:

```
>>> c
[(1, 0.5), (2, 1.0), (3, 1.5), (4, 2.0)]
>>> d=[i+j for i,j in c]
>>> d
[1.5, 3.0, 4.5, 6.0]
```

חיפוש איבר בעזרת פונקציה `.index()`.

לפונקציה הזאת יש 3 פרמטרים, כאשר פרמטר `item` הוא פרמטר חובה. `item` – זה איבר שאנו מחפשים. בנוסף יש 2 פרמטרים אופציונליים: `start` (מאיפה להתחיל חיפוש) ו-`end` (עד איזה איבר לבצע חיפוש).

```
tuple1 = (10, 11,12, 13, 14, 15, 16, 17, 18, 19, 20)
position = tuple1.index(16)
print(position)
```

6

אם ניתן איזור חיפוש לא מתאים או ערך שלא נמצא בצירוף, נקבל הודעת שגיעה:

```
tuple1 = (10, 11,12, 13, 14, 15, 16, 17, 18, 19, 20)
position = tuple1.index(30)
print(position)
position = tuple1.index(16,7,9)
print(position)
```

```
Traceback (most recent call last):
  File "C:\Users\marka\PycharmProjects\Game\main.py", line 2, in <module>
    position = tuple1.index(30)
ValueError: tuple.index(x): x not in tuple
```

```
Traceback (most recent call last):
  File "C:\Users\marka\PycharmProjects\Game\main.py", line 4, in <module>
    position = tuple1.index(16,7,9)
ValueError: tuple.index(x): x not in tuple
```

מה קורה כאשר יש יותר מערך 1 מתאים לחיפוש? חסרון של הפקודה שהיא מחזירה ערך ראשון שמתאים לקריטריון.

```
tuple1 = (20, 10, 11,12, 20, 13, 14, 15,20, 16, 17, 18, 19, 20)
position = tuple1.index(20)
```

```

print(position)
position = tuple1.index(20,2)
print(position)
position = tuple1.index(20,6,9)
print(position)

```

```

0
4
8

```

אם נרצה לדעת האם איבר נמצא בתוך צירוף נשתמש בפקודה in ונוכל לקבל תשובה true או false.  
אם נרצה לדעת כמה פעמים מופיע איבר בתוך צירוף, נשתמש בפקודה count.

קוד	הסבר
<pre> tuple1 = (10, 20, 60, 30, 60, 40, 60) print(600 in tuple1) print(60 in tuple1) count = tuple1.count(60) print(count) count = tuple1.count(600) print(count) </pre>	<p>הגדרת צירוף</p> <p>בדיקה האם מספר נמצא בתוך צירוף תוצאה ראשונה FALSE, שנייה - TRUE. ספירה כמה פעמים מופיע מספר 60 3 ספירה כמה פעמים מופיע מספר 600 0</p>

## תרגול בנושא צירופים.

תרגול בנושא טבלאות מעקב.  
תרגיל מס' 1.

תפענח את הקוד. תן דוגמה לפלט בהתאם לקלט שתבחר.

```

print("Enter numbers or n to finish")
x = i = 0
lst = []
while x != 'n':
    x = input()
    if (x != 'n'):
        lst.insert(i, int(x))
    i += 1
print(lst)
tuplex = tuple(lst)
print(tuplex)

```

תרגיל מס' 2.

תפענח את הקוד. מהו פלט של התוכנית?

```
tuple1 = (10, 20, 30, 40, 50)
print(tuple1)
tuple1 = tuple1[::-1]
print(tuple1)
```

תרגיל מס' 3.

תפענח את הקוד. תן דוגמה לפלט בהתאם לקלט שתבחר.

```
def try_convert(s):
    try:
        return int(s)
    except ValueError:
        try:
            return float(s)
        except ValueError:
            return s
print("Enter numbers or n to finish")
x = i = 0
lst = []
while x != 'n':
    x = input()
    if (x != 'n'):
        x = try_convert(x)
        lst.insert(i, x)
    i += 1
print(lst)
tuplex = tuple(lst)
print(tuplex)
```

תרגיל מס' 4.

תפענח את הקוד. תן דוגמה לפלט בהתאם לקלט שתבחר.

```
import random
print("Enter numbers or n to finish")
x = i = 0
lst = []
while x != 'n':
    x = input()
    if (x != 'n'):
```

```

lst.insert(i,int(x))
i+=1
print(lst)
tuplex = tuple(lst)
print(tuplex)
a=len(tuplex)
print(tuplex[random.randint(1,a)])

```

תרגיל מס' 5.

תפענח את הקוד. תן דוגמה לפלט בהתאם לקלט שתבחר.

```

def try_convert(s):
    try:
        if type(eval(s)) == float:
            return float(s)
        elif type(eval(x)) == int:
            return int(s)
        elif type(eval(x)) == bool:
            return bool(s)
    except:
        return s
print("Enter numbers or n to finish")
x = i = 0
lst = []
while x != 'n':
    x = input()
    if (x != 'n'):
        x = try_convert(x)
        lst.insert(i,x)
    i += 1
print(lst)
tuplex = tuple(lst)
print(tuplex)

```

תרגיל מס' 6.

תפענח את הקוד. מהו פלט של הקוד?

```

tuplex = (4, 8, 3)
print(tuplex)
n1, n2, n3 = tuplex
print(n1 + n2 + n3)

```

תרגיל מס' 7.

תפענח את הקוד. מהו פלט של הקוד?

```
tuplex = (4, 6, 2, 8, 3, 1)
print(tuplex)
tuplex = tuplex + (9,)
print(tuplex)
tuplex = tuplex[:5] + (15, 20, 25) + tuplex[:5]
print(tuplex)
listx = list(tuplex)
listx.append(30)
tuplex = tuple(listx)
print(tuplex)
```

תרגיל מס' 8.

תפענח את הקוד. מהו פלט של הקוד?

```
tup = ('e', 'x', 'e', 'r', 'c', 'i', 's', 'e', 's')
str = ".join(tup)
print(str)
```

תרגיל מס' 9.

תפענח את הקוד. מהו פלט של הקוד?

```
tuplex = ("w", "w", "w", ".", "e", "l", "e", "c", "s", "t", "u", "d", "y", ".", "o", "r", "g")
print(tuplex)
item = tuplex[3]
print(item)
item1 = tuplex[-4]
print(item1)
```

תרגיל מס' 10.

תפענח את הקוד. תן דוגמה לפלט בהתאם לקלט שתבחר.

```
tuplex = (2, 4, 5, 6, 2, 3, 4, 4, 7)
print(tuplex)
x=int(input())
count = tuplex.count(x)
print(count)
```

תרגיל מס' 11.

תפענח את הקוד. תן דוגמה לפלט בהתאם לקלט שתבחר.

```
print("Enter values for seek or space bar to exit")
s=0
tuplex = ("w", "w", "w", ".", "e", "l", "e", "c", "s", "t", "u", "d", "y", ".", "o", "r", "g")
while s!=" ":
    s=input()
    if s!=' ':
        print(s in tuplex)
print("Good bye")
```

תרגיל מס' 12.

תפענח את הקוד. מהו פלט של הקוד?

```
listx = [5, 10, 7, 4, 15, 3]
print(listx)
tuplex = tuple(listx)
print(tuplex)
```

תרגיל מס' 13.

תפענח את הקוד. מהו פלט של הקוד?

```
tuplex = ("w", "w", "w", ".", "e", "l", "e", "c", "s", "t", "u", "d", "y", ".", "o", "r", "g")
print(tuplex)
tuplex = tuplex[:2] + tuplex[3:]
print(tuplex)
listx = list(tuplex)
listx.remove("c")
tuplex = tuple(listx)
print(tuplex)
```

תרגיל מס' 14.

תפענח את הקוד. מהו פלט של הקוד?

```
tuplex = (2, 4, 3, 5, 4, 6, 7, 8, 6, 1)
slice = tuplex[3:5]
print(slice)
slice = tuplex[:6]
print(slice)
slice = tuplex[5:]
```

```

print(slice)
slice = tuplex[:]
print(slice)
slice = tuplex[-8:-4]
print(slice)
tuplex = tuple("HELLO WORLD")
print(tuplex)
slice = tuplex[2:9:2]
print(slice)
slice = tuplex[::4]
print(slice)
slice = tuplex[9:2:-4]
print(slice)

```

תרגיל מס' 15.

תפענח את הקוד. מהו פלט של הקוד?

```

tuplex = tuple("index tuple")
print(tuplex)
index = tuplex.index("p")
print(index)
index = tuplex.index("p", 5)
print(index)
index = tuplex.index("e", 3, 6)
print(index)

```

תרגיל מס' 16.

תפענח את הקוד. מהו פלט של הקוד?

```

tuplex = tuple("www.elecstudy.org")
print(tuplex)
print(len(tuplex))

```

תרגיל מס' 17.

תפענח את הקוד. מהו פלט של הקוד?

```

tuplex = ((2, "w"),(3, "r"))
print(dict((y, x) for x, y in tuplex))

```



תרגיל מס' 18.

תפענח את הקוד. מהו פלט של הקוד?

```
l = [(1,2), (3,4), (8,9)]
print(list(zip(*l)))
```

תרגיל מס' 19.

תפענח את הקוד. מהו פלט של הקוד?

```
x = ("www.elecstudy.org")
y = reversed(x)
print(tuple(y))
x = (5, 10, 15, 20)
y = reversed(x)
print(tuple(y))
```

תרגיל מס' 20.

תפענח את הקוד. מהו פלט של הקוד?

```
l = [("x", 1), ("x", 2), ("x", 3), ("y", 1), ("y", 2), ("z", 1)]
d = {}
for a, b in l:
    d.setdefault(a, []).append(b)
print(d)
```

תרגיל מס' 21.

תפענח את הקוד. מהו פלט של הקוד?

```
l = [(10, 20, 40), (40, 50, 60), (70, 80, 90)]
print(l)
print([t[:-1] + (100,) for t in l])
```

תרגיל מס' 22.

תפענח את הקוד. מהו פלט של הקוד?

```
L = [(), (), (','), ('a', 'b'), ('a', 'b', 'c'), ('d')]
L = [t for t in L if t]
print(L)
```

תרגיל מס' 23.

תפענח את הקוד. מהו פלט של הקוד?

```
price = [('item1', '12.20'), ('item2', '15.10'), ('item3', '24.5')]
print(price)
print( sorted(price, key=lambda x: float(x[1]), reverse=True))
```

תרגיל מס' 24.

תפענח את הקוד. מהו פלט של הקוד?

```
num = [10,20,30,(10,20),40]
ctr = 0
for n in num:
    if isinstance(n, tuple):
        break
    ctr += 1
print(ctr)
```

תרגול בנושא כתיבת קוד.

תרגיל מס' 1.

כתוב תוכנית ליצירת צירוף.

תרגיל מס' 2.

כתוב תוכנית ליצירת צירוף עם נתונים מסוגים שונים.

תרגיל מס' 3.

כתוב תוכנית ליצירת צירוף עם מספרים ותדפיס אחד מהם.

תרגיל מס' 4.

כתוב תוכנית שמעבירה איברים מצירוף למספר משתנים.

תרגיל מס' 5.

כתוב תוכנית שמוסיפה איברים לצירוף הקיים.

תרגיל מס' 6.

כתוב תוכנית שמעבירה צירוף למחרוזת.

תרגיל מס' 7.

כתוב תוכנית שמציגה איבר 4 מהתחלה ואיבר 4 מסוף הצירוף.

תרגיל מס' 8.

כתוב תוכנית שמחפשת איברים כפולים של צירוף.

תרגיל מס' 9.

כתוב תוכנית שבודקת האם איבר נמצא בתוך הצירוף.

תרגיל מס' 10.

כתוב תוכנית שהופכת רשימה לצירוף.

תרגיל מס' 11.

כתוב תוכנית למחיקת איבר מהצירוף.

תרגיל מס' 12.

כתוב תוכנית שחותכת את הצירוף.

תרגיל מס' 13.

כתוב תוכנית שמחפשת את המיקום של איבר בתוך הצירוף.

תרגיל מס' 14.

כתוב תוכנית לחיפוש אורך של הצירוף.

תרגיל מס' 15.

כתוב תוכנית להפיכת צירוף למילון. (בשלב מאוחר יותר שתלמד מה זה מילון).

תרגיל מס' 16.

כתוב תוכנית להפיכת רשימה של צירופים לרשימות נפרדות.

תרגיל מס' 17.

כתוב תוכנית להפיכת סדר של איברים בתוך הצירוף.

תרגיל מס' 18.

כתוב תוכנית להפיכת רשימה של צירופים למילון. (בשלב מאוחר יותר שתלמד מה זה מילון).

תרגיל מס' 19.

כתוב תוכנית להחלפת איבר אחרון בכל אחד מצירופים שנמצאות בתוך הרשימה. לדוגמה: רשימה מקורית: [(10, 20, 40), (40, 50, 60), (70, 80, 90)], רשימה לאחר שינוי: [(10, 20, 100), (40, 50, 100), (70, 80, 100)]

תרגיל מס' 20.

כתוב תוכנית שמורידה צירופים ריקים מרשימת הצירופים. לדוגמה - רשימה מקורית: [(), ('), ('d'), ('c', 'b', 'a'), ('c', 'b', 'a'), ('d', 'c', 'b', 'a')].

תרגיל מס' 21.

נתונה רשימת הצירופים, כאשר במקום ראשון של צירוף עומדת מחרוזת ובמקום שני מספר עם נקודה עשרונית. יש לסדר את הרשימה בסדר עולה לפי מספר עם נקודה עשרונית.

תרגיל מס' 22.

כתוב תוכנית שסופרת את כמות האיברים ברשימה כל עוד האיבר הוא לא מסוג צירוף.

## פרק י"א. מילון.

בשפת התכנות פייתון, מילונים מסוג "dict" הם סוג אחר של מבנה נתונים יחד עם רשימות וצירוף. מילון הוא מבנה נתונים שניתן לשינוי ועריכה (כמו רשימה) שאינו מסודר (בניגוד למחרוזות, רשימות וצירוף) של אלמנטים בעלי **מפתח : ערך**.

"לא מסודר" - פירושו שרצף הזוגות אינו חשוב. שפת התכנות לא לוקחת אותה בחשבון, וכתוצאה מכך אי אפשר לגשת לאלמנטים באמצעות אינדקסים.

בשפות אחרות מבנים הדומים למילונים נקראים באופן שונה. לדוגמה, ב-Java סוג נתונים כזה נקרא "מיפוי".

כדי להפוך את רעיון המילון למובן יותר, נשתמש באנלוגיה למילון קונבנציונאלי. למשל, אנגלית - עברית. לכל מילה באנגלית במילון כזה יש תרגום למילה בעברית: cat - חתול, dog - כלב, table - שולחן. אם מתואר המילון האנגלית-עברית באמצעות פייתון, ניתן ליצור מילים באנגלית ועברית -

המשמעות שלהם: {'cat': 'חתול', 'dog': 'כלב', 'bird': 'ציפור', 'mouse': 'עכבר'}

שימו לב לסוגריים מסולסלות {} - בעזרתם מוגדר מילון.

```
>>> a={"horse":"סוס", "dog":"כלב", "cat": "חתול"}
>>> a
{'horse': 'סוס', 'dog': 'כלב', 'cat': 'חתול'}
>>> a['cat']
'חתול'
>>> type(a)
<class 'dict'>
```

בדוגמה שלנו **מפתחות** הם: horse, dog, cat **וערכים (או משמעות)** הם: חתול, כלב, סוס.

מילון (כמו רשימות) ניתן לשינוי, וקיימות עבורו פעולות מחיקה והוספה של אלמנטים (יותר נכון זוגות של אלמנטים). ניתן ליצור בהתחלה מילון ריק: d={} ורק אחר כך להוסיף לו זוגות. כדי למחוק זוג מהמילון ניתן להשתמש בפקודה del ששייכת לקבוצת פעולות על מילון.

```
>>> a['elephant']='פיל'
>>> a
{'horse': 'סוס', 'dog': 'כלב', 'cat': 'חתול', 'elephant': 'פיל'}
>>> a['bird']='עכבראש'
>>> a['bird']='ציפור'
>>> a
{'horse': 'סוס', 'dog': 'כלב', 'cat': 'חתול', 'elephant': 'פיל', 'bird': 'ציפור'}
>>> del a['bird']
>>> a
{'horse': 'סוס', 'dog': 'כלב', 'cat': 'חתול', 'elephant': 'פיל'}
```

כדי לראות את כל האופרטורים של מילון נשתמש בפקודת `dir(dict)`.

```
>>> dir(dict)
['__class__', '__contains__', '__delattr__', '__delitem__', '__dir__', '__doc__', '__eq__',
 '__format__', '__ge__', '__getattr__', '__getitem__', '__gt__', '__hash__', '__init__',
 '__init_subclass__', '__iter__', '__le__', '__len__', '__lt__', '__ne__', '__new__', '__reduce__',
 '__reduce_ex__', '__repr__', '__reversed__', '__setattr__', '__setitem__', '__sizeof__', '__str__',
 '__subclasshook__', 'clear', 'copy', 'fromkeys', 'get', 'items', 'keys', 'pop', 'popitem', 'setdefault',
 'update', 'values']
```

## אופרטורים ששייכים לקבוצה של מילון הם:

```
'clear', 'copy', 'fromkeys', 'get', 'items', 'keys', 'pop', 'popitem', 'setdefault', 'update', 'values'
```

שימו לב שלא יכול להיות מצב שבאותו מילון יש שני זוגות עם אותו **מפתח**, אבל כן יכול להיות מצב שלשני **מפתחות** יהיו **ערכים** זהים.

מפתח יכול להיות כל דבר – מספר, מחרוזת, מספר עם נקודה עשרונית וכו'. ניתן לערבב אותם באותו מילון.

```
>>> nums = {1: 'one', 2: 'two', 3: 'three'}
>>> nums
{1: 'one', 2: 'two', 3: 'three'}
>>> person = {'name': 'Tom', 1: [30, 15, 16], 2: 2.34, ('ab', 100): 'no'}
>>> person
{'name': 'Tom', 1: [30, 15, 16], 2: 2.34, ('ab', 100): 'no'}
>>> nums[1]
'one'
>>> person[1]
[30, 15, 16]
>>> person[('ab',100)]
'no'
```

נראה דוגמה של הדפסת איברים בעזרת משפט `for`.

```
>>> nums
{1: 'one', 2: 'two', 3: 'three'}
>>> for i in nums:
    print(i)
1
2
3
>>> for i in nums:
    print(nums[i])
one
```

```
two
three
```

שימו לב שבברירת מחדל i מייצג את המפתח ו- nums[i] מייצג את הערך. ישנו אופרטור items() שבונה מבנה מיוחד שבנוי מצירוף. כל צירוף מכיל מפתח וערך:

```
>>> n = nums.items()
>>> n
dict_items([(1, 'one'), (2, 'two'), (3, 'three')])
```

בלולאה ניתן לפתוח את צירוף כדי שתהיה אפשרות להוציא גם מפתח וגם ערך:

```
>>> for key, value in nums.items():
    print(key, 'is', value)
1 is one
2 is two
3 is three
```

ניתן להשתמש באופרטורים keys(), values() כדי לקבל בנפרד מפתחות וערכים.

```
>>> v_nums = []
>>> for v in nums.values():
    v_nums.append(v)
>>> v_nums
['one', 'two', 'three']
```

## אופרטורים נוספים של מחלקת מילון.

אופרטור	הסבר
clear()	מנקה את התוכן של המילון, אבל לא את המילון עצמו – כתוצאה מקבלים מילון ריק.
copy()	שומר עותק למילון במקום אחר, כדי לעשות גיבוי נתונים.
fromkeys()	בונה מילון מאיברים של רשימה, כאשר איברים של רשימה מועברים בתור מפתחות.
get()	נוהג ערך של איבר של מילון לפי מפתח.

```
>>> nums
{1: 'one', 2: 'two', 3: 'three'}
>>> nums.clear()
>>> nums
{}
>>> a
{'horse': 'סוס', 'dog': 'כלב', 'cat': 'חתול', 'elephant': 'פיל'}
>>> b=a.copy()
>>> b
{'horse': 'סוס', 'dog': 'כלב', 'cat': 'חתול', 'elephant': 'פיל'}
>>> c=[1,2,3,4,5]
>>> d=dict.fromkeys(c)
>>> d
{1: None, 2: None, 3: None, 4: None, 5: None}
>>> c
```

```
[1, 2, 3, 4, 5]
>>> b.get("horse")
'סוס'
```

הסבר	אופרטור
מוחק זוג מהמילון לפי מפתח ומחזיר ערך של הזוג הנמחק.	pop()
מוחק זוג <b>אקראי</b> מהמילון ומחזיר ערך של הזוג הנמחק.	popitem()
מוסיף איברים למילון. זה דומה לפעולה רגילה של העתקה בעזרת אופרטור "=". ההבדל שכאשר משתמשים באופרטור "=" ואיבר קיים, אופרטור זה מחליף את הערך הישן בערך החדש. פעולה setdefault() לא משנה ערך הקיים.	setdefault()
מוסיף מילון נוסף למילון אחר.	update()

```
>>> b.pop("horse")
'סוס'
>>> b
{'dog': 'כלב', 'cat': 'חתול', 'elephant': 'פיל'}
>>> e=b.pop("cat")
>>> e
'חתול'
>>> b
{'dog': 'כלב', 'elephant': 'פיל'}
>>> e=b.popitem()
>>> e
('elephant', 'פיל')
>>> b
{'dog': 'כלב'}
>>> b.setdefault("cat","חתול")
'חתול'
>>> b
{'dog': 'כלב', 'cat': 'חתול'}
>>> e=b.setdefault("horse","סוס")
>>> b
{'dog': 'כלב', 'cat': 'חתול', 'horse': 'סוס'}
>>> e
'סוס'
>>> b["dog"]="dog"
>>> b
{'dog': 'dog', 'cat': 'חתול', 'horse': 'סוס'}
>>> b.setdefault("dog","כלב")
'dog'
>>> b
{'dog': 'dog', 'cat': 'חתול', 'horse': 'סוס'}
```



תרגול בנושא מילונים:  
תרגול בנושא טבלאות מעקב.  
תרגיל מס' 1.

תפענח את הקוד. מהו פלט של התוכנית?

```
a = [1, 2, 3, 4, 5]
b = ['a', 'b', 'c', 'd', 'e']
c = {}
if(len(a)<len(b)):
    length=len(a)
else:
    length=len(b)
for i in range(length):
    key = b[i]
    value= a[i]
    c[key] = value
print(c)
```

תרגיל מס' 2.

תפענח את הקוד. מהו פלט של התוכנית?

```
a = [1, 2, 3, 4, 5]
b = ['a', 'b', 'c']
c = {}
if(len(a)<len(b)):
    length=len(a)
else:
    length=len(b)
for i in range(length):
    key = b[i]
    value= a[i]
    c[key] = value
print(c)
```

תרגיל מס' 3.

תפענח את הקוד. מהו פלט של התוכנית?

```
a = [1, 2, 3, 4, 5]
b = ['a', 'b', 'c']
c = {}
c = dict(zip(b, a))
print(c)
```

תרגיל מס' 4.

תפענח את הקוד. מהו פלט של התוכנית?

```
import random
d = {"A": 4, "O": 6, "P": 10, "M": 7, "B": 3}
print(d)
keys = list(d.keys())
what_del = random.choice(keys)
del d[what_del]
print(d)
```

תרגיל מס' 5.

תפענח את הקוד. מהו פלט של התוכנית?

```
d = {'a': 10, 'e': 3, 'c': 4, 'd': 20, 'b': 15}
print("Sorting by keys:")
list_keys = list(d.keys())
list_keys.sort()
for i in list_keys:
    print(i, ':', d[i], end="; ")
print()
print("Sorting by values")
list_d = list(d.items())
list_d.sort(key=lambda i: i[1])
for i in list_d:
    print(i[0], ':', i[1], end="; ")
```

תרגיל מס' 6.

תפענח את הקוד. תן דוגמה לפלט בהתאם לקלט שתבחר.

```
goods = {"Apple": 4.5, "Orange": 6.2,
         "Pineapple": 10.0,
         "Mango": 7.5,
         "Banana": 3.8}
for good, price in goods.items():
    print(good, " - ", price, end="; ")
print()
cost = 0
while True:
    good = input("What? (n - nothing) ")
    if good == 'n':
        break
    qty = int(input("How many? "))
    cost += goods[good] * qty
print("Price:", cost)
```

תרגיל מס' 7.

תפענח את הקוד. מהו פלט של התוכנית?

```
import operator
d = {1: 2, 3: 4, 4: 3, 2: 1, 0: 0}
print('Original dictionary : ',d)
sorted_d = dict(sorted(d.items(), key=operator.itemgetter(1)))
print('Dictionary in ascending order by value : ',sorted_d)
sorted_d = dict(sorted(d.items(), key=operator.itemgetter(1),reverse=True))
print('Dictionary in descending order by value : ',sorted_d)
```

תרגיל מס' 8.

תפענח את הקוד. תן דוגמה לפלט בהתאם לקלט שתבחר.

```
d = {0:10, 1:20}
print(d)
a=int(input())
b=int(input())
d.update({a:b})
print(d)
```

תרגיל מס' 9.

תפענח את הקוד. מהו פלט של התוכנית?

```
dic1={1:10, 2:20, 5:50}
dic2={3:30, 4:40,6:60}
dic3={}
for d in (dic1, dic2): dic3.update(d)
print(dic3)
```

תרגיל מס' 10.

תפענח את הקוד. מהו פלט של התוכנית?

```
d = {1: 10, 2: 20, 3: 30, 4: 40, 5: 50, 6: 60}
def is_key_present(x):
    if x in d:
        print('Key is present in the dictionary')
    else:
        print('Key is not present in the dictionary')
is_key_present(5)
is_key_present(9)
```

תרגיל מס' 11.

תפענח את הקוד. מהו פלט של התוכנית? תן דוגמה לפלט בהתאם לקלט שתבחר.

```
d = {'x': 10, 'y': 20, 'z': 30}
for dict_key, dict_value in d.items():
    print(dict_key,'->',dict_value)
```

תרגיל מס' 12.

תפענח את הקוד. תן דוגמה לפלט בהתאם לקלט שתבחר.

```
n=int(input("Input a number "))
d = dict()
for x in range(1,n+1):
    d[x]=x*x
print(d)
```

תרגיל מס' 13.

תפענח את הקוד. מהו פלט של התוכנית?

```
my_dict = {'data1':100,'data2':-54,'data3':247}
print(sum(my_dict.values()))
```

תרגיל מס' 14.

תפענח את הקוד. מהו פלט של התוכנית?

```
my_dict = {'data1':100,'data2':-54,'data3':247}
result=1
for key in my_dict:
    result=result * my_dict[key]
print(result)
```

תרגיל מס' 15.

תפענח את הקוד. מהו פלט של התוכנית?

```
myDict = {'a':1,'b':2,'c':3,'d':4}
print(myDict)
if 'a' in myDict:
    del myDict['a']
print(myDict)
```

תרגיל מס' 16.

תפענח את הקוד. מהו פלט של התוכנית?

```
keys = ['red', 'green', 'blue']
values = ['#FF0000', '#008000', '#0000FF']
color_dictionary = dict(zip(keys, values))
print(color_dictionary)
```

תרגיל מס' 17.

תפענח את הקוד. מהו פלט של התוכנית?

```
color_dict = {'red': '#FF0000',
              'green': '#008000',
              'black': '#000000',
              'white': '#FFFFFF'}
for key in sorted(color_dict):
    print("%s: %s" % (key, color_dict[key]))
```

תרגיל מס' 18.

תפענח את הקוד. מהו פלט של התוכנית?

```
my_dict = {'x': 500, 'y': 5874, 'z': 560}
key_max = max(my_dict.keys(), key=(lambda k: my_dict[k]))
key_min = min(my_dict.keys(), key=(lambda k: my_dict[k]))
print('Maximum Value: ', my_dict[key_max])
print('Minimum Value: ', my_dict[key_min])
```

תרגיל מס' 19.

תפענח את הקוד. מהו פלט של התוכנית?

```
student_data = {'id1':
                {'name': ['Sara'],
                 'class': ['V'],
                 'subject_integration': ['english, math, science']
                },
                'id2':
                {'name': ['David'],
                 'class': ['V'],
                 'subject_integration': ['english, math, science']
                },
                'id3':
```

```

{'name': ['Sara'],
 'class': ['V'],
 'subject_integration': ['english, math, science']}
},
'id4':
{'name': ['Surya'],
 'class': ['V'],
 'subject_integration': ['english, math, science']}
},
}
print()
result = {}
print(student_data)

for key,value in student_data.items():
    if value not in result.values():
        result[key] = value
print(result)

```

תרגיל מס' 20.

תפענח את הקוד. מהו פלט של התוכנית?

```

my_dict = {}
if not bool(my_dict):
    print("Dictionary is empty")

```

תרגיל מס' 21.

תפענח את הקוד. מהו פלט של התוכנית?

```

from collections import Counter
d1 = {'a': 100, 'b': 200, 'c':300}
d2 = {'a': 300, 'b': 200, 'd':400}
d = Counter(d1) + Counter(d2)
print(d)

```

תרגיל מס' 22.

תפענח את הקוד. מהו פלט של התוכנית? תן דוגמה לפלט בהתאם לקלט שתבחר.

```

L = [{"V": "S001"}, {"V": "S002"}, {"VI": "S001"}, {"VI": "S005"}, {"VII": "S005"},
 {"V": "S009"}, {"VIII": "S007"}]
print("Original List: ",L)
u_value = set( val for dic in L for val in dic.values())
print("Unique Values: ",u_value)

```

תרגיל מס' 23.

תפענח את הקוד. מהו פלט של התוכנית?

```
import itertools
d={'1':['a','b'], '2':['c','d']}
for combo in itertools.product(*[d[k] for k in sorted(d.keys())]):
    print("".join(combo))
```

תרגיל מס' 24.

תפענח את הקוד. מהו פלט של התוכנית?

```
from heapq import nlargest
my_dict = {'a':500, 'b':5874, 'c': 560,'d':400, 'e':5874, 'f': 20}
print(my_dict)
three_largest = nlargest(3, my_dict, key=my_dict.get)
print(three_largest)
```

תרגיל מס' 25.

תפענח את הקוד. מהו פלט של התוכנית?

```
from collections import defaultdict, Counter
str1 = 'www.elecstudy.org'
my_dict = {}
for letter in str1:
    my_dict[letter] = my_dict.get(letter, 0) + 1
print(my_dict)
```

תרגיל מס' 26.

תפענח את הקוד. מהו פלט של התוכנית?

```
my_dict = {'C1':[1,2,3], 'C2':[5,6,7], 'C3':[9,10,11]}
for row in zip(*([key] + (value) for key, value in sorted(my_dict.items()))):
    print(*row)
```

תרגיל מס' 27.

תפענח את הקוד. מהו פלט של התוכנית?

```
student = [{'id': 1, 'success': True, 'name': 'Lary'},
            {'id': 2, 'success': False, 'name': 'Rabi'},
            {'id': 3, 'success': True, 'name': 'Alex'}]
print(sum(d['success'] for d in student))
```

תרגיל מס' 28.

תפענח את הקוד. מהו פלט של התוכנית?

```
num = {'n1': [2, 3, 1], 'n2': [5, 1, 2], 'n3': [3, 2, 4]}
sorted_dict = {x: sorted(y) for x, y in num.items()}
print(sorted_dict)
```

תרגיל מס' 29.

תפענח את הקוד. מהו פלט של התוכנית?

```
student_list = {'S 001': ['Math', 'Science'], 'S 002': ['Math', 'English']}
print("Original dictionary: ", student_list)
student_dict = {x.translate({'32': None}): y for x, y in student_list.items()}
print("New dictionary: ", student_dict)
```

תרגיל מס' 30.

תפענח את הקוד. מהו פלט של התוכנית?

```
from heapq import nlargest
from operator import itemgetter
items = {'item1': 45.50, 'item2': 35, 'item3': 41.30, 'item4': 55, 'item5': 24}
for name, value in nlargest(3, items.items(), key=itemgetter(1)):
    print(name, value)
```

תרגיל מס' 31.

תפענח את הקוד. מהו פלט של התוכנית?

```
students = {'Aex':{'class':'V',
    'roll_id':2},
    'Puja':{'class':'V',
    'roll_id':3}}
for a in students:
    print(a)
    for b in students[a]:
        print (b,':',students[a][b])
```



תרגיל מס' 32.

תפענח את הקוד. מהו פלט של התוכנית?

```
student = {
    'name': 'Alex',
    'class': 'V',
    'roll_id': '2'
}
print(student.keys() >= {'class', 'name'})
print(student.keys() >= {'name', 'Alex'})
print(student.keys() >= {'roll_id', 'name'})
```

תרגיל מס' 33.

תפענח את הקוד. מהו פלט של התוכנית?

```
dict = {'Alex': ['subj1', 'subj2', 'subj3'], 'David': ['subj1', 'subj2']}
ctr = sum(map(len, dict.values()))
print(ctr)
```

תרגיל מס' 34.

תפענח את הקוד. מהו פלט של התוכנית?

```
from collections import Counter
x = Counter({'Math':81, 'Physics':83, 'Chemistry':87})
print(x.most_common())
```

תרגיל מס' 35.

תפענח את הקוד. מהו פלט של התוכנית?

```
def sum_math_v_vi_average(list_of_dicts):
    for d in list_of_dicts:
        n1 = d.pop('V')
        n2 = d.pop('VI')
        d['V+VI'] = (n1 + n2)/2
    return list_of_dicts
student_details= [
    {'id': 1, 'subject': 'math', 'V': 70, 'VI': 82},
    {'id': 2, 'subject': 'math', 'V': 73, 'VI': 74},
    {'id': 3, 'subject': 'math', 'V': 75, 'VI': 86}
]
print(sum_math_v_vi_average(student_details))
```

תרגיל מס' 36.

תפענח את הקוד. מהו פלט של התוכנית?

```
x = {'key1': 1, 'key2': 3, 'key3': 2}
y = {'key1': 1, 'key2': 2}
for (key, value) in set(x.items()) & set(y.items()):
    print('%s: %s is present in both x and y' % (key, value))
```

תרגיל מס' 37.

תפענח את הקוד. מהו פלט של התוכנית?

```
from pprint import pprint
dict_nums = dict(x=list(range(11, 20)), y=list(range(21, 30)), z=list(range(31, 40)))
pprint(dict_nums)
print(dict_nums["x"][4])
print(dict_nums["y"][4])
print(dict_nums["z"][4])
for k,v in dict_nums.items():
    print(k, "has value", v)
```

תרגיל מס' 38.

תפענח את הקוד. מהו פלט של התוכנית?

```
dict1 = {'c1': 'Red', 'c2': 'Green', 'c3': None}
print("Original Dictionary:")
print(dict1)
print("New Dictionary after dropping empty items:")
dict1 = {key:value for (key, value) in dict1.items() if value is not None}
print(dict1)
```

תרגיל מס' 39.

תפענח את הקוד. מהו פלט של התוכנית?

```
marks = {'Cierra Vega': 175, 'Alden Cantrell': 180, 'Kierra Gentry': 165, 'Pierre Cox': 190}
print("Original Dictionary:")
print(marks)
print("Marks greater than 170:")
result = {key:value for (key, value) in marks.items() if value >= 170}
print(result)
```

## תרגול בנושא כתיבת קוד.

תרגיל מס' 1.

תייצר 2 רשימות. תבנה מילון, כאשר ה"מפתחות" הם איברים של רשימה ראשונה וה"ערכים" הם איברים של רשימה השנייה.

מספר החזרות צריך להיות לא יותר גדול מהרשימה הקצרה ביותר במידה ורשימות באורך שונה.

תרגיל מס' 1א.

תפתור תרגיל מס' 1 בשימוש בפונקציה בנויה בשם `zip()`. פונקציה שלוקחת צירופים שבנויים משני איברים. פונקציה `dict()` באופן אוטומטי לוקחת רשימות ובונה מהם מילון.

תרגיל מס' 2.

נתון מילון. כתוב תוכנית שמוחקת מהמילון זוג באקראי.

שיטת `keys()` בונה אובייקט שמכיל מפתחות של המילון. ניתן לבנות ממנו רשימה של מפתחות בעזרת שיטת `list()`. פונקציה `choice()` בוחרת אובייקט מהרשימה ומעתיקה את המיקום שלו למשתנה עזר. בעזרת שיטה `del` ניתן למחוק אותו מהמילון.

תרגיל מס' 3.

נתון מילון, כאשר ה"מפתחות" הם מחרוזות, ו"ערכים" הם מספרים. תדפיס אותו לפי סדר א-ב של מפתחות ואז לפי סדר עולה של ערכים.

למעשה, לא ניתן למיין את תוכן מילון, מכיוון שמילון בפייתון הוא מבנה נתונים לא מסודר. גם אם תוסיפו פריטים למילון בצורה מסודרת, כמו למשל באופן אלפביתי, הם עשויים להופיע אחרת כשמוצגים. עם זאת, בעת אחזור פריטים ממילון, ניתן לגרום לזה לקרות בסדר ספציפי. לשם כך נעשה שימוש במבנה מסודר שניתן למיין. לדוגמה, רשימה.

הדרך הקלה ביותר היא למיין את המילון לפי מקשים. אלגוריתם ליציאת תוכן המילון:

1. צור רשימת מפתחות מילון.

2. מיין.

3. בלולאה `for`, חזור על רכיבי הרשימה באמצעות אלמנט הרשימה כמפתח המילון.

מיון מילון לפי ערכים קשה יותר, מכיוון שניתן לגשת לאלמנטים של מילון רק על ידי "מפתחות". עם זאת, ניתן ליצור רשימת צירופים ("מפתח", "ערך") ולמיין אותה לפי האלמנט השני של הזוגות. בהמשך התוכנית משמש המבנה המסודר הזה ולא המילון המקורי עצמו.

תרגיל מס' 4.

נתון מילון עם מפתחות (שמות של מצרכים) וערכים (מחירים שלהם). כתוב תוכנית שמדפיסה על המסך את הנתונים ושואלת את המשתמש איזה מוצר הוא רוצה לקנות כמה מצרכים מכל סוג הוא רוצה לקנות. התוכנית מחשבת את המחיר של העסקה ומדפיסה אותו.

שיטת items() מחזירה אובייקט מסוג צירוף שמכיל "מפתח" – "ערך". בכל סיבוב מוציאים צירוף הבא.

תרגיל מס' 5.

כתוב תוכנית שממיינת מילון לפי ערך.

תרגיל מס' 6.

כתוב תוכנית להוספת איבר למילון.

תרגיל מס' 7.

כתוב תוכנית שמקבלת 2 מילונים ומאחדת אותם למילון משותף.

תרגיל מס' 8.

כתוב תוכנית שבודקת האם איבר קיים במילון או לא.

תרגיל מס' 9.

כתוב תוכנית שעוברת דרך מילון בעזרת לולאת for ומדפיסה תוכן של מילון.

תרגיל מס' 10.

כתוב תוכנית שמייצרת ומדפיסה מילון שמכיל מספרים (בין 1 לאין סוף) במקום ראשון וריבוע שלו במקום שני. תוכנית מקבלת מהמשתמש מספר איברים של המילון ומדפיסה את המילון.

תרגיל מס' 11.

כתוב תוכנית שמחברת את כל הערכים של מילון.

תרגיל מס' 12.

כתוב תוכנית שמחשבת כפל של כל הערכים של המילון.

תרגיל מס' 13.

כתוב תוכנית שמורידה איבר מהמילון.

תרגיל מס' 14.

כתוב תוכנית שמקבלת 2 רשימות ומחזירה מילון שבנוי מהם.

תרגיל מס' 15.

כתוב תוכנית למיון מילון לפי מפתח.

תרגיל מס' 16.

כתוב תוכנית למציאת ערכים הקטן ביותר והגדול ביותר של המילון.

תרגיל מס' 17.

כתוב תוכנית למציאת כפילויות מהמילון.

תרגיל מס' 18.

כתוב תוכנית שבודקת האם מילון ריק או לא.

תרגיל מס' 19.

כתוב תוכנית שמחברת 2 מילונים. במקרה וקיים אותו מפתח בשניהם יש לחבר את הערכים שלהם.

לדוגמה:  $d2 = \{ 'a': 300, 'b': 200, 'd': 400 \}$   $d1 = \{ 'a': 100, 'b': 200, 'c': 300 \}$

תוצאה:  $\{ 'a': 400, 'b': 400, 'd': 400, 'c': 300 \}$

תרגיל מס' 20.

כתוב תוכנית שמדפיסה רק ערכים ייחודיים של המילון.

תרגיל מס' 21.

תבנה מילון שהמפתח שלו הוא מספר וערך בנוי מרשימה עם שתי אותיות. כתוב תוכנית שמציגה את כל האפשרויות של אותיות האלה. לדוגמה: מילון  $\{ '1': ['a', 'b'], '2': ['c', 'd'] \}$  פלט: bd bc ad ac

תרגיל מס' 22.

כתוב תוכנית לחיפוש 3 ערכים הגדולים ביותר במילון.

תרגיל מס' 23.

כתוב תוכנית שבונה מילון ממחרוזת כך, שהמפתח הוא אות של המחרוזת והערך הוא כמות הפעמים שאות מופיעה במחרוזת.

תרגיל מס' 24.

כתוב תוכנית שמדפיסה את המילון בצורת טבלה.

תרגיל מס' 25.

נתונה רשימת מילונים שמורכבת מ-3 שדות: ת.ז. הצלחה במבחן ושם. לדוגמה:  
Sample data: = [{ 'id': 1, 'success': True, 'name': 'Lary' }, { 'id': 2, 'success': False, 'name': 'Rabi' }, { 'id': 3, 'success': True, 'name': 'Alex' }]  
בדוק כמה אנשים עברו את המבחן וכמה לא.

תרגיל מס' 26.

כתוב תוכנית שמקבלת מילון, כאשר ערכים של המילון הם רשימות. לדוגמה:  
num = { 'n1': [2, 3, 1], 'n2': [5, 1, 2], 'n3': [3, 2, 4] }  
תוכנית צריכה למיין ערכים של רשימות מהקטן לגדול.

תרגיל מס' 27.

כתוב תוכנית להורדת רווחים של מפתחות מהמילון.

Original dictionary: { 'S 001': ['Math', 'Science'], 'S 002': ['Math', 'English'] }

New dictionary: { 'S001': ['Math', 'Science'], 'S002': ['Math', 'English'] }

תרגיל מס' 28.

תבנה מילון שמתאר חנות, כאשר המפתח הוא סוג המוצר והערך הוא מחיר. כתוב תוכנית שמציגה 3 מוצרים היקרים ביותר עם המחירים שלהם.

תרגיל מס' 29.

כתוב תוכנית שמדפיסה מילון שורה אחרי שורה.

תרגיל מס' 30.

כתוב תוכנית שמחפשת מפתחות כפולים במילון.

תרגיל מס' 31.

כתוב תוכנית שמכילה מילון, כאשר המפתח הוא שם והערך היא רשימת מקוצעות שתלמיד לומד. תוכנית צריכה לחשב כמה איברים יש בכל רשימה.

תרגיל מס' 32.

כתוב תוכנית שממיינת את המילון בהתאם לערך. לדוגמה:

Sample data : {'Math':81, 'Physics':83, 'Chemistry':87}

Expected data: [('Chemistry', 87), ('Physics', 83), ('Math', 81)]

תרגיל מס' 33.

כתוב תוכנית שמחליפה במילון שני ציונים בממוצע שלהם.

תרגיל מס' 34.

כתוב תוכנית שמקבלת 2 מילונים ובודקת איזה מפתח וערך מופיע גם במילון ראשון וגם בשני. יש להדפיס אותו.

תרגיל מס' 35.

הגדר מילון עם 3 מפתחות X, Y, Z. למפתח ראשון ערך הוא רשימת הערכים שלמים מ-11 עד 19, למפתח שני מ-21 עד 29 ולמפתח השלישי ערכים מ-31 עד 39.

תרגיל מס' 36.

כתוב תוכנית שבודקת את המילון ומוחקת משם את כל האיברים ללא ערכים.

תרגיל מס' 37.

כתוב תוכנית כדי לסנן איברים של מילון בהתאם לקריטריון שמסנן ערכים שלו.

## פרק י"ב. סטים. Sets

### מה זה סט?

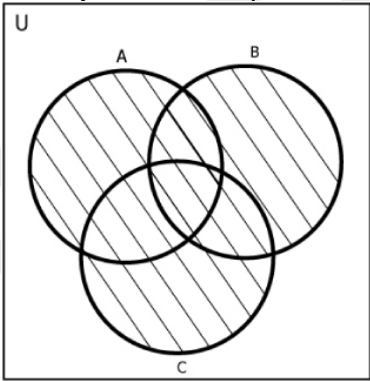
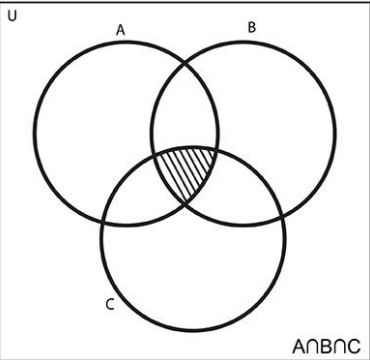
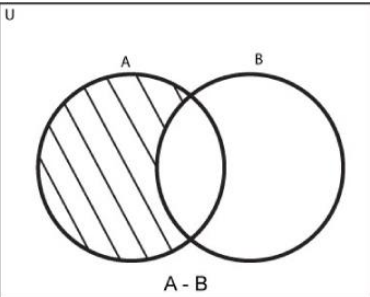
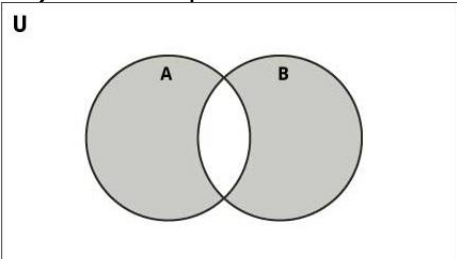
סט או קבוצה הוא מבנה נתונים מורכב נוסף של פייתון המובנה בקטגוריית האוסף, יחד עם רשימות, מילונים וכו'. הסט הוא מפוזר כמו מילון, ניתן לשינוי כמו מילון ורשימה, הוא פשוט מכיל אלמנטים כרשימה, ולא מפתח: זוג ערכים כמילון. מאפיין הוא הייחודיות של כל אחד מהיסודות שלו. במילים אחרות, במילון יכולים להיות שני אלמנטים עם ערך של נניח 100, אך בסט זה לא יכול להיות. מושג הסטים הגיע מהמתמטיקה. כמו כן, רעיון הסטים משמש בגרפיקה ממוחשבת. ישנן שלוש פעולות בסיסיות המבוצעות על סטים. איחוד, חיתוך והפרש. נניח שיש לך שתי דמויות שכל נקודה מתוארת על ידי קואורדינטות. כל דמות נקבעת על פי הנקודות הרבות שלה. כאשר משולבים דמויות אלה, כל הנקודות של שתי הדמויות ייפלו לסט, אך אם הדמויות חופפות, אז הנקודות של אזור החפיפה בערכה המתקבלת יוצגו ביחיד. במקרה של חיתוך, רק הנקודות שיש לדמות הראשונה והן לשנייה במערכת התוצאות ייפלו. במילים אחרות, אזור החפיפה שלהם. כשאתה מוצא את ההבדל בין הסטים, זה משנה ממה מופרע. הסט הכולל כולל את כל הנקודות של "המפורסמות", למעט נקודות המשותפות ל"חסר". בשפת התכנות פייתון, ניתן ליצור מערכת הניתנת להחלפה בשתי דרכים. הראשון, בסוגריים מסולסלות רשמו את האלמנטים המופרדים בפסיקים. שנית, התקשר לפונקציה המובנה .set()

שימו לב שבסט a "1" לא חוזר על עצמו. לפונקציה set() אפשר לשלוח רק את האובייקט ולא את רשימת האיברים העצמם.

```
>>> a={1,2,3,4,1}
>>> a
{1, 2, 3, 4}
>>> b=[1,2,3]
>>> b=set(b)
>>> b
{1, 2, 3}
>>> c=set(1,2,3)
Traceback (most recent call last):
  File "<pyshell#59>", line 1, in <module>
    c=set(1,2,3)
TypeError: set expected at most 1 argument, got 3
>>> set("abc")
{'b', 'c', 'a'}
>>> set([1,2,4])
{1, 2, 4}
```



## אופרטורים בין סטים הן:

הסבר	אופרטור
$s1$ זהה ל- $s2$	$s1 == s2$
$s1$ שונה מ- $s2$	$s1 != s2$
$s1$ תת סט של- $s2$	$s1 \leq s2$
$s1$ תת סט של $s2$	$s1 < s2$
$s1$ מכיל את $s2$	$s1 \geq s2$
$s1$ מכיל את $s2$	$s1 > s2$
חיבור בין שני סטים (איחוד).	$s1 \cup s2$
	
חיתוך בין שני סטים (מה שמשותף בין שני סטים).	$s1 \cap s2$
	
איברים של סט $s1$ שלא נמצאים בסט $s2$	$s1 - s2$
	
פעולת XOR לוגי בין 2 סטים. (זה כמו איחוד של סטים, לא כולל זוגות זהות).	$s1 \oplus s2$
	

```
>>> a={(1,2), (3,4), (2,0), (0,1)}
>>> b={(0,0), (0,1),(2,0),(0,2), (1,1)}
>>> a|b
{(0, 0), (1, 1), (0, 2), (2, 0), (1, 2), (3, 4), (0, 1)}
>>> a&b
{(0, 1), (2, 0)}
>>> a-b
{(1, 2), (3, 4)}
>>> a^b
{(0, 0), (1, 2), (1, 1), (3, 4), (0, 2)}
>>> a={1,2,3,4,5,6,7,8,9,10}
>>> b={0,2,4,6,8,10}
>>> c={1,3,5,7,9,11}
>>> e=set.copy(a)
>>> e
{1, 2, 3, 4, 5, 6, 7, 8, 9, 10}
>>> a==e
True
>>> a==b
False
>>> a!=b
True
>>> a!=e
False
>>> a>=c
False
>>> a>c
False
>>> f={1,3,5}
>>> a>=f
True
>>> a>f
True
```

## ניתן לבצע את אותם הפעולות בעזרת פונקציות מיוחדות:

a.union(b) - |

a.intersection(b) - &

a.difference(b) -

a.symmetric\_difference(b) ^

```
>>> a={(1,2), (3,4), (2,0), (0,1)}
>>> b={(0,0), (0,1),(2,0),(0,2), (1,1)}
>>> a.union(b)
{(0, 0), (1, 1), (0, 2), (2, 0), (1, 2), (3, 4), (0, 1)}
>>> a.intersection(b)
{(0, 1), (2, 0)}
>>> a.difference(b)
{(1, 2), (3, 4)}
>>> a.symmetric_difference(b)
{(0, 0), (1, 2), (1, 1), (3, 4), (0, 2)}
```

ישנה פקודה in שמאפשרת לדעת האם האובייקט נמצא בסט או לא:

```
>>> a={(1,2), (3,4), (2,0), (0,1)}
>>> (0,0) in a
False
>>> (1,2) in a
True
>>> (0,0) not in a
True
>>> (1,2) not in a
False
```

כדי לראות את כל הפקודות של סטים נשתמש בפקודה .dir(set).

```
>>> dir(set)
['_and_', '_class_', '_contains_', '_delattr_', '_dir_', '_doc_', '_eq_', '_format_',
'_ge_', '_getattr_', '_gt_', '_hash_', '_iand_', '_init_', '_init_subclass_',
'_ior_', '_isub_', '_iter_', '_ixor_', '_le_', '_len_', '_lt_', '_ne_', '_new_',
'_or_', '_rand_', '_reduce_', '_reduce_ex_', '_repr_', '_ror_', '_rsub_', '_rxor_',
'_setattr_', '_sizeof_', '_str_', '_sub_', '_subclasshook_', '_xor_', 'add', 'clear',
'copy', 'difference', 'difference_update', 'discard', 'intersection', 'intersection_update', 'isdisjoint',
'issubset', 'issuperset', 'pop', 'remove', 'symmetric_difference', 'symmetric_difference_update',
'union', 'update']
```

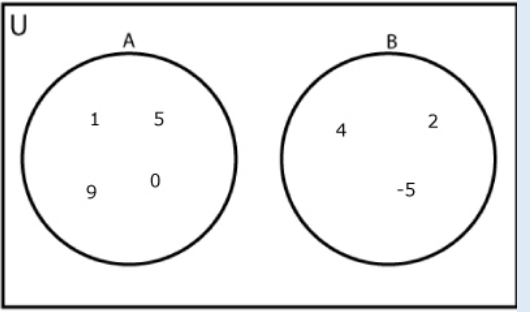
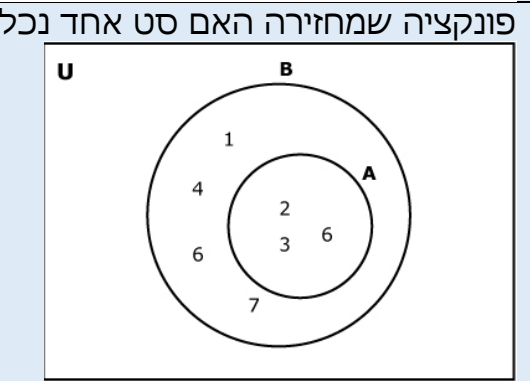
## חוץ מפונקציות שראינו ישנם עוד מספר פונקציות בנוגע לסטים:

פונקציה	פירוש
add	הוספה של איבר חדש לסט הקיים. אם האיבר קיים הפונקציה לא מוסיפה אותו פעם נוספת.
clear	ניקוי התוכן של הסט הקיים ללא מחיקה שלו. יתקבל סט ריק.
copy	מעתיקה את התוכן של הסט הקיים לסט אחר (קיים או חדש).
discard	הוצאת איבר מהסט, כאשר מציינים שם באיבר, אם האיבר לא קיים בסט הפונקציה לא תבצע כלום <b>ולא מחזירה את החריגה.</b>
pop	הוצאת איבר אקראי מסט
remove	הוצאת איבר מהסט, כאשר מציינים שם באיבר, אם האיבר לא קיים בסט פונקציה <b>מחזירה את החריגה.</b>

```
>>> a={(1,2), (3,4), (2,0), (0,1)}
>>> set.add(a,(0,0))
>>> a
{(0, 0), (2, 0), (1, 2), (3, 4), (0, 1)}
>>> set.clear(a)
>>> a
set()
>>> a={(1,2), (3,4), (2,0), (0,1)}
>>> b=set.copy(a)
>>> b
{(1, 2), (3, 4), (0, 1), (2, 0)}
>>> set.discard(b,(0,1))
>>> b
{(1, 2), (3, 4), (2, 0)}
>>> set.pop(a)
(1, 2)
>>> a
{(3, 4), (0, 1), (2, 0)}
>>> set.remove(a,(2,0))
>>> a
{(3, 4), (0, 1)}
```

## פונקציות נוספות בנוגע לסטים.

פונקציה	פירוש
update	הוספת איברים לסט מסטים אחרים
difference_update	הכנסת הפרש של 2 סטים לסט המקורי (סט שאותו מחסירים לא משתנה).
intersection_update	הוספת חיתוך של 2 סטים לסט המקורי (הסט השני לא משתנה).
symmetric_difference_update	הכנסת תוצאת XOR של 2 סטים לסט המקורי (הסט השני לא משתנה).
isdisjoint	פונקציה שמחזירה השוואה ומציגה האם שני סטים משותפים או לא. מספיק שיש לפחות אלמנט אחד משותף.

		
פונקציה שמחזירה האם סט אחד נכלל בתוך סט שני.		issubset
פונקציה שמחזירה האם סט אחד מכיל את הסט השני.		issuperset

```

>>> A = {1, 2, 3, 4}
>>> B = {5, 6, 7}
>>> C = {4, 5, 6}
>>> print('Are A and B disjoint?', A.isdisjoint(B))
Are A and B disjoint? True
>>> print('Are A and C disjoint?', A.isdisjoint(C))
Are A and C disjoint? False
>>> A = {1, 2, 3}
>>> B = {1, 2, 3, 4, 5}
>>> C = {1, 2, 4, 5}
>>> print(A.issubset(B))
True
>>> print(A.issubset(C))
False
>>> print(C.issubset(B))
True

```

## דוגמה לעבודה עם סטים:

```

set1 = set() # Creating two sets
set2 = set()

for i in range(1, 6): # Adding elements to set1
    set1.add(i)

```

```

for i in range(3, 8): # Adding elements to set2
    set2.add(i)

print("Set1 = ", set1)
print("Set2 = ", set2)

set3 = set1 | set2 # set1.union(set2) # Union of set1 and set2
print("Union of Set1 & Set2: Set3 = ", set3)

set4 = set1 & set2 # set1.intersection(set2) # Intersection of set1 and set2
print("Intersection of Set1 & Set2: Set4 = ", set4)

# Checking relation between set3 and set4
if set3 > set4: # set3.issuperset(set4)
    print("Set3 is superset of Set4")
elif set3 < set4: # set3.issubset(set4)
    print("Set3 is subset of Set4")
else: # set3 == set4
    print("Set3 is same as Set4")

# displaying relation between set4 and set3
if set4 < set3: # set4.issubset(set3)
    print("Set4 is subset of Set3")

set5 = set3 - set4 # difference between set3 and set4
print("Elements in Set3 and not in Set4: Set5 = ", set5)
print("\n")

if set4.isdisjoint(set5): # checkv if set4 and set5 are disjoint sets
    print("Set4 and Set5 have nothing in common\n")

set5.clear() # Removing all the values of set5

print("After applying clear on sets Set5: ")
print("Set5 = ", set5)

```

```

Set1 = {1, 2, 3, 4, 5}
Set2 = {3, 4, 5, 6, 7}

Union of Set1 & Set2: Set3 = {1, 2, 3, 4, 5, 6, 7}
Intersection of Set1 & Set2: Set4 = {3, 4, 5}

Set3 is superset of Set4
Set4 is subset of Set3

```

Elements in Set3 and not in Set4: Set5 = {1, 2, 6, 7}

Set4 and Set5 have nothing in common

After applying clear on sets Set5:

Set5 = set()

טסליצקי מרק

תרגילים בנושא סטים:  
תרגול בנושא טבלאות מעקב.  
תרגיל מס' 1.

תפענח את הקוד. מהו פלט של התוכנית?

```
x = set()
print(x)
n = set([0, 1, 2, 3, 4])
print(n)
```

תרגיל מס' 2.

תפענח את הקוד. מהו פלט של התוכנית?

```
num_set = set([0, 1, 2, 3, 4, 5])
for n in num_set:
    print(n)
```

תרגיל מס' 3.

תפענח את הקוד. מהו פלט של התוכנית?

```
color_set = set()
color_set.add("Red")
print(color_set)
color_set.update(["Blue", "Green"])
print(color_set)
```

תרגיל מס' 4.

תפענח את הקוד. מהו פלט של התוכנית?

```
num_set = set([0, 1, 3, 4, 5])
num_set.pop()
print(num_set)
num_set.pop()
print(num_set)
```

תרגיל מס' 5.

תפענח את הקוד. מהו פלט של התוכנית?

```
num_set = set([0, 1, 2, 3, 4, 5])
print(num_set)
num_set.discard(4)
print(num_set)
```



תרגיל מס' 6.

תפענח את הקוד. מהו פלט של התוכנית?

```
setx = set(["green", "blue", "red", "yellow"])
sety = set(["blue", "yellow", "orange"])
setz = setx & sety
print(setz)
```

תרגיל מס' 7.

תפענח את הקוד. מהו פלט של התוכנית?

```
setx = set(["green", "blue", "red", "yellow"])
sety = set(["blue", "yellow", "orange"])
setz = setx | sety
print(setz)
```

תרגיל מס' 8.

תפענח את הקוד. מהו פלט של התוכנית?

```
setx = set(["green", "blue", "red", "yellow"])
sety = set(["blue", "yellow", "orange"])
setz = setx - sety
print(setz)
```

תרגיל מס' 9.

תפענח את הקוד. מהו פלט של התוכנית?

```
setx = set(["green", "blue", "red", "yellow"])
sety = set(["blue", "yellow", "orange"])
setz = setx ^ sety
print(setz)
```

תרגיל מס' 10.

תפענח את הקוד. מהו פלט של התוכנית?

```
print("Check if a set is a subset of another set, using comparison operators and issubset():\n")
setx = set(["apple", "mango"])
sety = set(["mango", "orange"])
setz = set(["mango"])
print("x: ", setx)
```

```
print("y: ",sety)
print("z: ",setz,"\n")
print("If x is subset of y")
print(setx <= sety)
print(setx.issubset(sety))
print("If y is subset of x")
print(sety <= setx)
print(sety.issubset(setx))
print("\nIf y is subset of z")
print(sety <= setz)
print(sety.issubset(setz))
print("If z is subset of y")
print(setz <= sety)
print(setz.issubset(sety))
```

תרגיל מס' 11.

תפענח את הקוד. מהו פלט של התוכנית?

```
setp = set(["Red", "Green"])
setq = setp.copy()
print(setq)
setq.clear()
print(setq)
```

תרגיל מס' 12.

תפענח את הקוד. מהו פלט של התוכנית?

```
seta = set([5, 10, 3, 15, 2, 20])
print(max(seta))
print(min(seta))
```

תרגיל מס' 13.

תפענח את הקוד. מהו פלט של התוכנית?

```
seta = set([5, 10, 3, 15, 2, 20])
print(len(seta))
```

תרגיל מס' 14.

תפענח את הקוד. מהו פלט של התוכנית?

```
set1 = set()
print("Initial blank Set: ")
print(set1)

set1 = set("www.elecstudy.org")
print("\n Set with the use of String: ")
print(set1)

String = ' www.elecstudy.org '
set1 = set(String)
print("\n Set with the use of an Object: " )
print(set1)

set1 = set(["www ", " elecstudy ", " org "])
print("\n Set with the use of List: ")
print(set1)
```

תרגיל מס' 15.

תפענח את הקוד. מהו פלט של התוכנית?

```
set1 = set([1, 2, 4, 4, 3, 3, 3, 6, 5])
print("\nSet with the use of Numbers: ")
print(set1)
set1 = set([1, 2, 'Geeks', 4, 'For', 6, 'Geeks'])
print("\nSet with the use of Mixed Values")
print(set1)
```

תרגיל מס' 16.

תפענח את הקוד. מהו פלט של התוכנית?

```
set1 = set()
print("Initial blank Set: ")
print(set1)
set1.add(8)
set1.add(9)
set1.add((6,7))
print("\nSet after Addition of Three elements: ")
print(set1)
for i in range(1, 6):
    set1.add(i)
print("\nSet after Addition of elements from 1-5: ")
print(set1)
```

תרגיל מס' 17.

תפענח את הקוד. מהו פלט של התוכנית?

```
set1 = set([ 4, 5, (6, 7)])
set1.update([10, 11])
print("\nSet after Addition of elements using Update: ")
print(set1)
```

תרגיל מס' 18.

תפענח את הקוד. מהו פלט של התוכנית?

```
set1 = set(["www", "elecstudy", "org"])
print("\nInitial set")
print(set1)
print("\nElements of set: ")
for i in set1:
    print(i, end=" ")
print("www" in set1)
```

תרגיל מס' 19.

תפענח את הקוד. מהו פלט של התוכנית?

```
set1 = set([1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12])
print("Initial Set: ")
print(set1)
set1.remove(5)
set1.remove(6)
print("\nSet after Removal of two elements: ")
print(set1)
set1.discard(8)
set1.discard(9)
print("\nSet after Discarding two elements: ")
print(set1)
for i in range(1, 5):
    set1.remove(i)
print("\nSet after Removing a range of elements: ")
print(set1)
```

תרגיל מס' 20.

תפענח את הקוד. מהו פלט של התוכנית?

```
def vowel_count(str):
    count = 0
    vowel = set("aeiouAEIOU")
    for alphabet in str:
        if alphabet in vowel:
            count = count + 1
    print("No. of vowels :", count)
str = "www.elecstusy.org"
vowel_count(str)
```

תרגיל מס' 21.

תפענח את הקוד. תן דוגמה לפלט בהתאם לקלט שתבחר.

```
from string import ascii_lowercase as asc_lower
def check(s):
    return set(asc_lower) - set(s.lower()) == set([])
string = "The quick brown fox jumps over the lazy dog"
if(check(string)== True):
    print("The string is a pangram")
else:
    print("The string isn't a pangram")
```

תרגיל מס' 22.

תפענח את הקוד. מהו פלט של התוכנית?

```
num_set = set([0, 1, 2, 3, 4, 5])
for n in num_set:
    print(n, end=' ')
print("\n Creating a set using string:")
char_set = set("www.elecstudy.org")
for val in char_set:
    print(val, end=' ')
```

תרגיל מס' 23.

תפענח את הקוד. מהו פלט של התוכנית?

```
num_set = set([0, 1, 2, 3, 4, 5])
print("Original set elements:")
print(num_set)
print("\nRemove 0 from the said set:")
num_set.discard(0)
print(num_set)
print("\nRemove 5 from the said set:")
num_set.discard(5)
print(num_set)
print("\nRemove 2 from the said set:")
num_set.discard(2)
print(num_set)
print("\nRemove 7 from the said set:")
num_set.discard(7)
print(num_set)
```

תרגיל מס' 24.

תפענח את הקוד. מהו פלט של התוכנית?

```
sn1 = {1,2,3,4,5}
sn2 = {4,5,6,7,8}
print("Original sets:")
print(sn1)
print(sn2)
print("\nRemove the intersection of a 2nd set from the 1st set using difference_update():")
sn1.difference_update(sn2)
print("sn1: ",sn1)
print("sn2: ",sn2)
sn1 = {1,2,3,4,5}
sn2 = {4,5,6,7,8}
print("\nRemove the intersection of a 2nd set from the 1st set using -= operator:")
sn1-=sn2
print("sn1: ",sn1)
print("sn2: ",sn2)
```

תרגיל מס' 25.

תפענח את הקוד. מהו פלט של התוכנית?

```
evodSet = {78, 11, 54, 95, 16, 36, 61, 77, 150, 122}
print("Even and Odd Set Items = ", evodSet)
sEvenCount = sOddCount = 0
for eoVal in evodSet:
    if(eoVal % 2 == 0):
        sEvenCount = sEvenCount + 1
    else:
        sOddCount = sOddCount + 1
print("The Count of Even Numbers in evodSet = ", sEvenCount)
print("The Count of Odd Numbers in evodSet = ", sOddCount)
```

תרגיל מס' 26.

תפענח את הקוד. תן דוגמה לפלט בהתאם לקלט שתבחר.

```
evodSet = set()
number = int(input("Enter the Total Even Odd Set Items = "))
for i in range(1, number + 1):
    value = int(input("Enter the %d Set Item = " %i))
    evodSet.add(value)
print("Even and Odd Set Items = ", evodSet)
sEvenCount = sOddCount = 0
for eoVal in evodSet:
    if(eoVal % 2 == 0):
        sEvenCount = sEvenCount + 1
    else:
        sOddCount = sOddCount + 1
print("The Count of Even Numbers in evodSet = ", sEvenCount)
print("The Count of Odd Numbers in evodSet = ", sOddCount)
```

תרגיל מס' 27.

תפענח את הקוד. תן דוגמה לפלט בהתאם לקלט שתבחר.

```
def CountOfSetEvenandOddNumbers(evodSet):
    sEvenCount = sOddCount = 0
    for eoVal in evodSet:
        if(eoVal % 2 == 0):
            sEvenCount = sEvenCount + 1
        else:
            sOddCount = sOddCount + 1
```

```

    return sEvenCount, sOddCount
evodSet = set()
number = int(input("Enter the Total Even Odd Set Items = "))
for i in range(1, number + 1):
    value = int(input("Enter the %d Set Item = " %i))
    evodSet.add(value)
print("Even and Odd Set Items = ", evodSet)
sECount, sOCount = CountOfSetEvenandOddNumbers(evodSet)
print("The Count of Even Numbers in evodSet = ", sECount)
print("The Count of Odd Numbers in evodSet = ", sOCount)

```

תרגיל מס' 28.

תפענח את הקוד. מהו פלט של התוכנית?

```

pongSet = {6, -33, 99, -28, -56, 11, -45, -43}
print("Positive and Negative Set Items = ", pongSet)
sPositiveCount = sNegativeCount = 0
for eoVal in pongSet:
    if(eoVal >= 0):
        sPositiveCount = sPositiveCount + 1
    else:
        sNegativeCount = sNegativeCount + 1
print("The Count of Positive Numbers in pongSet = ", sPositiveCount)
print("The Count of Negative Numbers in pongSet = ", sNegativeCount)

```

תרגיל מס' 29.

תפענח את הקוד. תן דוגמה לפלט בהתאם לקלט שתבחר.

```

pongSet = set()
number = int(input("Enter the Total Positive Negative Set Items = "))
for i in range(1, number + 1):
    value = int(input("Enter the %d Set Item = " %i))
    pongSet.add(value)
print("Positive and Negative Set Items = ", pongSet)
sPositiveCount = sNegativeCount = 0
for eoVal in pongSet:
    if(eoVal >= 0):
        sPositiveCount = sPositiveCount + 1
    else:
        sNegativeCount = sNegativeCount + 1
print("The Count of Positive Numbers in pongSet = ", sPositiveCount)
print("The Count of Negative Numbers in pongSet = ", sNegativeCount)

```



תרגיל מס' 30.

תפענח את הקוד. תן דוגמה לפלט בהתאם לקלט שתבחר.

```
def CountOfSetPositiveandNegativeNumbers(pongSet):
    sPositiveCount = sNegativeCount = 0
    for eoVal in pongSet:
        if(eoVal >= 0):
            sPositiveCount = sPositiveCount + 1
        else:
            sNegativeCount = sNegativeCount + 1
    return sPositiveCount, sNegativeCount
pongSet = set()
number = int(input("Enter the Total Positive Negative Set Items = "))
for i in range(1, number + 1):
    value = int(input("Enter the %d Set Item = " %i))
    pongSet.add(value)
print("Positive and Negative Set Items = ", pongSet)
sECount, sOCount = CountOfSetPositiveandNegativeNumbers(pongSet)
print("The Count of Positive Numbers in pongSet = ", sECount)
print("The Count of Negative Numbers in pongSet = ", sOCount)
```

תרגיל מס' 31.

תפענח את הקוד. מהו פלט של התוכנית?

```
intSet = {10, 25, 136, 44, 150}
print("Set Items = ", intSet)
intSetLength = len(intSet)
print("Set Length = ", intSetLength)
strSet = {'apple', 'banana', 'Mango', 'kiwi'}
print("String Set Items = ", strSet)
strSetLength = len(strSet)
print("String Set Length = ", strSetLength)
mSet = {'Apple', 2.46, 45, 'Banana', (2, 4, 6, 8), 1}
print("Mixed Set Items = ", mSet)
mSetLength = len(mSet)
print("Mixed Set Length = ", mSetLength)
```

תרגיל מס' 32.

תפענח את הקוד. תן דוגמה לפלט בהתאם לקלט שתבחר.

```
intSet = set()
number = int(input("Enter the Total Set Items = "))
for i in range(1, number + 1):
    value = int(input("Enter the %d Set value = " %i))
    intSet.add(value)
print("Set Items = ", intSet)
intSetLength = len(intSet)
print("Set Length = ", intSetLength)
```

תרגיל מס' 33.

תפענח את הקוד. תן דוגמה לפלט בהתאם לקלט שתבחר.

```
mxSet = set()
number = int(input("Enter the Total Set Items = "))
for i in range(1, number + 1):
    value = int(input("Enter the %d Set Item = " %i))
    mxSet.add(value)
print("Set Items = ", mxSet)
sortedVal = sorted(mxSet)
print("Largest Item in mxSet Set = ", sortedVal[len(mxSet) - 1])
print("Data Type of sortedVal = ", type(sortedVal))
```

תרגיל מס' 34.

תפענח את הקוד. תן דוגמה לפלט בהתאם לקלט שתבחר.

```
def SetLargest(mxSet, setLargest):
    for i in mxSet:
        if(setLargest < i):
            setLargest = i
    return setLargest
mxSet = set()
number = int(input("Enter the Total Set Items = "))
for i in range(1, number + 1):
    value = int(input("Enter the %d Set Item = " %i))
    mxSet.add(value)
setLargest = value
print("Set Items = ", mxSet)
lar = SetLargest(mxSet, setLargest)
print("Largest Item in mxSet Set = ", lar)
```

תרגיל מס' 35.

תפענח את הקוד. מהו פלט של התוכנית?

```
evenoddSet = {78, 64, 11, 95, 36, 66, 77, 151}
print("Even and Odd Set Items = ", evenoddSet)
sEvenSum = sOddSum = 0
for eoVal in evenoddSet:
    if(eoVal % 2 == 0):
        sEvenSum = sEvenSum + eoVal
    else:
        sOddSum = sOddSum + eoVal
print("The Sum of Even Numbers in evenoddSet = ", sEvenSum)
print("The Sum of Odd Numbers in evenoddSet = ", sOddSum)
```

תרגיל מס' 36.

תפענח את הקוד. תן דוגמה לפלט בהתאם לקלט שתבחר.

```
evenoddSet = set()
number = int(input("Enter the Total Even Odd Set Items = "))
for i in range(1, number + 1):
    value = int(input("Enter the %d Set Item = " %i))
    evenoddSet.add(value)
print("Even and Odd Set Items = ", evenoddSet)
sEvenSum = sOddSum = 0
for eoVal in evenoddSet:
    if(eoVal % 2 == 0):
        sEvenSum = sEvenSum + eoVal
    else:
        sOddSum = sOddSum + eoVal
print("The Sum of Even Numbers in evenoddSet = ", sEvenSum)
print("The Sum of Odd Numbers in evenoddSet = ", sOddSum)
```

תרגיל מס' 37.

תפענח את הקוד. מהו פלט של התוכנית? תן דוגמה לפלט בהתאם לקלט שתבחר.

```
def sumOfSetEvenandOddNumbers(evenoddSet):
    sEvenSum = sOddSum = 0
    for eoVal in evenoddSet:
        if(eoVal % 2 == 0):
            sEvenSum = sEvenSum + eoVal
        else:
            sOddSum = sOddSum + eoVal
```

```
    return sEvenSum, sOddSum
evenoddSet = set()
number = int(input("Enter the Total Even Odd Set Items = "))
for i in range(1, number + 1):
    value = int(input("Enter the %d Set Item = " %i))
    evenoddSet.add(value)
print("Even and Odd Set Items = ", evenoddSet)
sESum, sOSum = sumOfSetEvenandOddNumbers(evenoddSet)
print("The Sum of Even Numbers in evenoddSet = ", sESum)
print("The Sum of Odd Numbers in evenoddSet = ", sOSum)
```

תרגול בנושא כתיבת קוד.

תרגיל מס' 1.

כתוב תוכנית ליצירת סט.

תרגיל מס' 2.

כתוב תוכנית שעוברת על איברים בתוך סט ומדפיסה אותם.

תרגיל מס' 3.

כתוב תוכנית להוספת איברים לתוך סט.

תרגיל מס' 4.

כתוב תוכנית להורדת איברים מסט.

תרגיל מס' 5.

כתוב תוכנית להורדת איברים מסט אם הם קיימים בתוך סט, לפי דוגמה הבאה.

{0, 1, 2, 3, 4, 5}  
{0, 1, 2, 3, 5}

תרגיל מס' 6.

כתוב תוכנית שבודקת חיתוך בין סטים.

תרגיל מס' 7.

כתוב תוכנית שבודקת איחוד בין סטים.

תרגיל מס' 8.

כתוב תוכנית שבודקת הפרש בין סטים.

תרגיל מס' 9.

כתוב תוכנית שבודקת פעולת XOR בין סטים.

תרגיל מס' 10.

כתוב תוכנית שבודקת פעולות issubset ו-issuperset לפי פלט הבא:

Check if a set is a subset of another set, using comparison operators and issubset():

```
x: {'apple', 'mango'}  
y: {'mango', 'orange'}  
z: {'mango'}
```

If x is subset of y

False

False

If y is subset of x

False

False

If y is subset of z

False

False

If z is subset of y

True

True

תרגיל מס' 11.

כתוב תוכנית שמנקה סט מאיברים.

תרגיל מס' 12.

כתוב תוכנית שמחפשת ערכים הקטן ביותר והגדול ביותר של הסט.

20

2

תרגיל מס' 13.

כתוב תוכנית שמחפשת אורך של סט (כמות איברים בתוך סט).

תרגיל מס' 14.

נתונה סט של 1000 מספרים אקראיים בתחום מ-1 עד 1000. קבע כמה מספרים שונים נמצאים בו ותדפיס אותם.

תרגיל מס' 15.

טסליצקי מרק

נתונה סט של 100 מספרים אקראיים בתחום מ-1 עד 100. עבור כל מספר, הדפס את המספר והמילה YES אם מספר זה התרחש בעבר בסט, או NO אם לא.

תרגיל מס' 16.

כתוב תוכנית שבודקת האם כל האותיות "תנועות" (aeiouy) קיימות בתוך המשפט הנקלט. (הגדר סט של "תנועות" מסודרות לפי א"ב וסט הנקלט. הפוך את כל האותיות של המשפט הנקלט לאותיות קטנות, תוציא ממנו את כל התנועות ותבדוק שכל התנועות קיימות.

לדוגמה:

**Input :** www.elecstudy.org

**Output :** Not all vowels are present

**Input :** AByeeIghiObhkUul

**Output :** All vowels are present

תרגיל מס' 17.

כתוב תוכנית שמקבלת 3 סטים ומוציאה איברים משותפים בין כולם.

לדוגמה:

קלט לדוגמה:

ar1 = [1, 5, 10, 20, 40, 80]

ar2 = [6, 7, 20, 80, 100]

ar3 = [3, 4, 15, 20, 30, 70, 80, 120]

פלט:

[80, 20]

תרגיל מס' 18.

נתון סט S של אותיות קטנות. המשימה היא לבדוק אם סט הנתון היא הטרוגרמה או לא. הטרוגרמה היא מילה, ביטוי או משפט שבהם לא מופיעה אות באלפבית יותר מפעם אחת.

לדוגמה:

**Input :** S = "the big dwarf only jumps"

**Output :** Yes

Each alphabet in the string S is occurred only once.

**Input :** S = "www.elecstudy.org"

**Output :** No

'w', '.', 'e' occurred more than once.

תרגיל מס' 19.

בדוק האם שני סטים מכילים לפחות איבר אחד משותף.  
לדוגמה:

Input : a = [1, 2, 3, 4, 5]

b = [5, 6, 7, 8, 9]

Output : True

Input : a=[1, 2, 3, 4, 5]

b=[6, 7, 8, 9]

Output : False

תרגיל מס' 20.

הגדר סט של 20 מספרים אקראיים בין 1 ל-10. תדפיס אותו. חשב סכום של כל המספרים שעומדים במקומות זוגיים וכל המספרים שעומדים במקומות אי-זוגיים והצג את הסכומים.

תרגיל מס' 21.

הגדר סט של 20 מספרים אקראיים בין 1 ל-10. תדפיס אותו. חשב סכום של כל המספרים זוגיים וכל המספרים אי-זוגיים והצג הסכומים.



## פרק י"ג'. קבצים.

### אחסון נתונים בתוך קבצים.

הגיון לאחסון כמויות גדולות של נתונים לא ברשימות או במילונים, אלא בקבצים. לכן שפות תכנות מספקות את היכולת לעבוד עם קבצים. בפייתון מטפלים בקבצים כמו באובייקטים. משמע, קובץ הוא אובייקט, כלומר קובץ טקסט הוא סוג נתונים יחד עם סוגי רשימה, מילון, מספר שלם וכו'.

בדרך כלל הקבצים מחולקים לשני סוגים: קבצי טקסט וקבצים בינאריים. סוג ראשון של הקבצים נחשב כמכיל נתוני תווים, מחרוזות. סוג שני הוא זרימת בתים. קריאת ביתים מידע, לדוגמה, נקראים קבצי תמונה (RGB – בית לכל צבע).

העבודה עם קבצים בינאריים היא קצת יותר מסובכת. לעתים קרובות הם מעובדים באמצעות מודולים מיוחדים של פייתון (צירופים, מבנים). בפרק זה נעסוק בטכניקות הבסיסיות לקריאה וכתובת קבצי טקסט.

בתחילת החוברת ראינו עבודה עם קבצים. כאן נראה עבודת קבצים לעומק.

### פונקציה open() – פתיחת קובץ.

פתיחת קובץ נעשית באמצעות הפונקציה open() – פונקציה המובנית של פייתון. בדרך כלל מועברים לה ארגומנט אחד או שניים. הראשון הוא שם הקובץ או השם עם הכתובת, אם הקובץ אינו נמצא בספרייה בה נמצא הסקריפט. הארגומנט השני הוא המצב בו נפתח הקובץ.

בדרך כלל משתמשים במצבי קריאה ('r') וכתובה ('w'). אם הקובץ פתוח במצב קריאה, אז הכתיבה אליו אינה אפשרית. ניתן לקרוא רק נתונים ממנו. אם הקובץ נפתח במצב כתיבה, ניתן רק לכתוב לו נתונים, אתה לא יכול לקרוא אותו.

אם הקובץ נפתח במצב 'w', אז כל הנתונים שהיו לפני כן נמחקים.

הקובץ הופך ריק. אם אין צורך למחוק את הנתונים הקיימים בקובץ, במקום להשתמש במצב ההקלטה, יש להשתמש במצב החלף ('a').

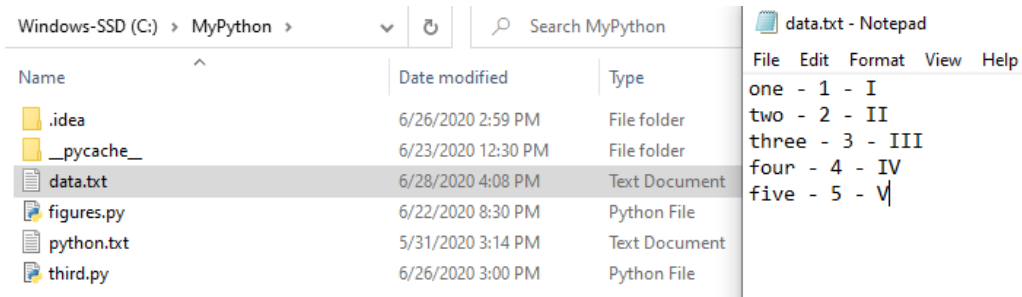
אם הקובץ חסר, פתיחתו במצב 'w' תיצור קובץ חדש. ישנם מצבים בהם יש צורך ליצור קובץ חדש, תוך הקפדה על הימנעות מחיקת נתונים קיימים בטעות. במקרה זה, במקום מצב 'w', נעשה שימוש במצב 'a'. זה תמיד יוצר קובץ חדש לכתובה. אם צוין שם קובץ קיים, ייזרק חריג, ואובדן נתונים בקובץ קיים לא יתרחש.

אם הארגומנט השני אינו מוגדר כאשר קוראים לפתוח (), הקובץ נפתח במצב קריאה כקובץ טקסט. לפתיחת הקובץ בתור בתים, התו 'b' מתווסף לאות של מצב הקריאה / הכתיבה. האות t מייצגת קובץ טקסט. מכיוון שזה סוג קובץ ברירת המחדל, הוא בדרך כלל לא מצוין. לא ניתן לציין רק את סוג הקובץ, כלומר פתיחת קובץ כך: "פתח"שם קובץ", 'b') תיתן שגיאה, גם אם הקובץ נפתח לקריאה. פתיחה נכונה תהיה כך: "פתח"שם קובץ", 'rb')". אנו יכולים לפתוח קבצי טקסט רק עם הפקודה פתח"שם קובץ"), מכיוון שגם 'r' וגם 't' משתמעים כברירת מחדל.

הפונקציה open() מחזירה אובייקט מסוג קובץ. עליו להיות משויך מיד למשתנה כדי לא להפסידו, או לקרוא מיד כארגומנט לפונקציה אחרת.

## קריאת תוכן הקובץ – פונקציה read().

בעזרת פונקציה read() ניתן לקרוא קובץ כולו או מספר בתים מסוים. נניח שיש לנו קובץ data.txt עם התוכן הבא:



נפתח אותו ונקרא:

```
>>> f1=open("C:\MyPython\data.txt")
>>> f1.read(10)
'one - 1 - '
>>> f1.read()
'I\ntwo - 2 - II\nthree - 3 - III\nfour - 4 - IV\nfive - 5 - V'
>>> f1.read()
''
>>> type(f1.read())
<class 'str'>
```

בהתחלה קולטים 10 בתים ראשונים – שזה בעצם 10 תווים. זה לא קובץ בינארי, אבל בכל זאת קולטים בית אחרי בית. קריאה הבאה (ללא פרמטרים) קוראת תווים עד סוף הקובץ. אחרי זה אובייקט f1 נשאר ריק. שימו לב שפונקציה read() מחזירה מחרוזת וסוף השורה נראית כך - 'n'.

## קריאת שורה מהקובץ – פונקציה readline().

```
>>> f1=open("C:\MyPython\data.txt")
>>> f1.readline()
'one - 1 - I\n'
>>> f1.readline()
'two - 2 - II\n'
>>> f1.readline()
'three - 3 - III\n'
>>> f1.readline()
'four - 4 - IV\n'
>>> f1.readline()
'five - 5 - V'
>>> f1.readline()
''

>>> f1=open("C:\MyPython\data.txt")
>>> f2=f1.readline()
>>> f2
'one - 1 - I\n'
```

## קריאת קובץ שורה לאחר שורה – פונקציה `.readlines()`

הפונקציה הזאת לא רק קוראת את הקובץ, אלה גם בונה רשימה. ברגע שמעתיקים רשימה למשתנה חדש – הוא נשמר שם וניתן לקרוא אותו אין סוף פעמים.

```
>>> f1=open("C:\MyPython\data.txt")
>>> f1.readlines()
['one - 1 - I\n', 'two - 2 - II\n', 'three - 3 - III\n', 'four - 4 - IV\n', 'five - 5 - V']
>>> f1=open("C:\MyPython\data.txt")
>>> f2=f1.readlines()
>>> f1
<_io.TextIOWrapper name='C:\\MyPython\\data.txt' mode='r' encoding='cp1252'>
>>> f2
['one - 1 - I\n', 'two - 2 - II\n', 'three - 3 - III\n', 'four - 4 - IV\n', 'five - 5 - V']
>>> f2
['one - 1 - I\n', 'two - 2 - II\n', 'three - 3 - III\n', 'four - 4 - IV\n', 'five - 5 - V']
```

ניתן לקרוא את תוכן הקובץ ישירות מהקובץ ללא פונקציות מיוחדות:

```
>>> for i in open('C:\MyPython\data.txt'):
    print(i)
one - 1 - I

two - 2 - II

three - 3 - III

four - 4 - IV

five - 5 - V
```

במקרה כזה לולאה מתקיימת כל עוד לא הגענו לסוף הקובץ. רואים כאן שורות ריקות – פונקציה `print()` הופכת סימן `'\n'` לשורה חדשה. כדי לפתור בעיה זאת נעשה שימוש בפונקציה `.append()` משתנה `i` מקבל שורה חדשה כל פעם. אנו לוקחים אותה מהתחלה עד האיבר לפני האחרון (שהוא סימן של שורה חדשה). שימו לב שסימן `'\n'` – זה סימן אחד ולא שני סימנים.

```
>>> nums=[]
>>> for i in open('C:\MyPython\data.txt'):
    nums.append(i[:-1])
>>> nums
['one - 1 - I', 'two - 2 - II', 'three - 3 - III', 'four - 4 - IV', 'five - 5 - ']
```

## כתיבה לקובץ – פונקציה write().

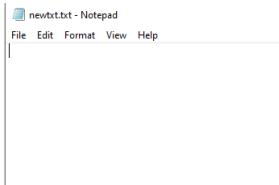
כתיבה לקובץ נעשה ע"י פונקציות write() ו-writelines().

פונקציה write() מעתיקה לקובץ נתון (חשוב מאוד שהוא יהיה מחרוזת או תו) ומחזירה את כמות התווים שהיא העתיקה לקובץ.

פונקציה writelines() מעתיקה לקובץ מבנה כמו רשימה, אבל לא מחזירה שום ערך.

```
>>> one=['1','2','3','4','5']
>>> one
['1','2','3','4','5']
>>> f2=open("c:\\MyPython\\newtxt.txt","w")
>>> f2.write("one")
3
>>> f2.write("two")
3
>>> f2.write("three")
5
>>> f2.write("four")
4
>>> f2.writelines(one)
```

כל עוד לא נסגור את הקובץ לא נוכל לראות שינויים שעשינו בתוך הקובץ.



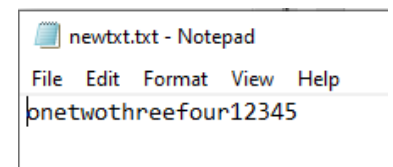
## סגירת קובץ – פונקציות close(), closed().

לאחר סגירת הקובץ נראה תמונה הבאה:

```
>>> f2.close()
>>> f2.closed
True
```

פקודת close() סוגרת את הקובץ ומשחררת אותו.

פקודה closed() מחזירה True אם הקובץ נסגר בצורה תקינה ו-False במקרה והקובץ נסגר בצורה לא תקינה.

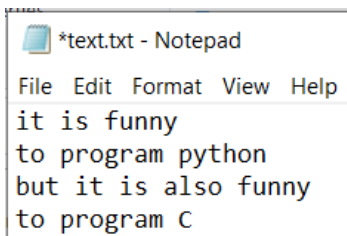


תרגול בנושא קבצים.  
תרגול בנושא טבלאות מעקב.  
תרגיל מס' 1.

תפענח את הקוד. מהו פלט של התוכנית?

```
data = []
f = open("C:\\MyPython\\text.txt")
for i in f:
    data.append(i)
print(data)
for i in range(len(data)):
    if data[i][-1] == '\n':
        data[i] = data[i][:-1]
print(data)
```

תוכן של הקובץ:



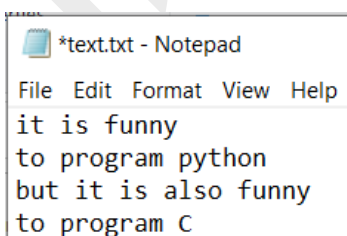
```
*text.txt - Notepad
File Edit Format View Help
it is funny
to program python
but it is also funny
to program C
```

תרגיל מס' 2.

תפענח את הקוד. מהו פלט של התוכנית?

```
data = []
f = open("C:\\MyPython\\text.txt")
for i in f:
    data.append(i)
print(data)
for i in range(len(data)):
    data[i] = data[i].strip()
print(data)
```

תוכן של הקובץ:



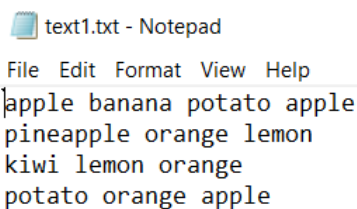
```
*text.txt - Notepad
File Edit Format View Help
it is funny
to program python
but it is also funny
to program C
```

### תרגיל מס' 3.

תפענח את הקוד. תן דוגמה לפלט בהתאם לקלט שתבחר ותוכן הקובץ הנתון.

```
f = open("C:\\MyPython\\text1.txt")
word = input()
word_count = 0
for line in f:
    word_list = line.split()
    word_count += word_list.count(word)
print(word_count)
```

תוכן של הקובץ:



text1.txt - Notepad

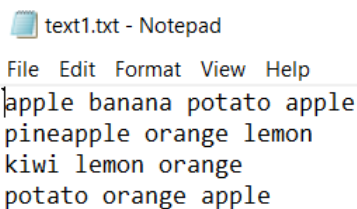
```
File Edit Format View Help
apple banana potato apple
pineapple orange lemon
kiwi lemon orange
potato orange apple
```

### תרגיל מס' 4.

תפענח את הקוד. תן דוגמה לפלט בהתאם לקלט שתבחר ותוכן הקובץ.

```
f = open("C:\\MyPython\\text1.txt")
all_lines = f.read()
word = input()
word_count = all_lines.count(word)
print(word_count)
```

תוכן של הקובץ:



text1.txt - Notepad

```
File Edit Format View Help
apple banana potato apple
pineapple orange lemon
kiwi lemon orange
potato orange apple
```

### תרגיל מס' 5.

תפענח את הקוד. מהו פלט של התוכנית בהתאם לתוכן הקובץ?

```
lines = 0
words = 0
letters = 0
for line in open("C:\\MyPython\\text1.txt"):
```

```

lines += 1
letters += len(line)
pos = 'out'
for letter in line:
    if letter != ' ' and pos == 'out':
        words += 1
        pos = 'in'
    elif letter == ' ':
        pos = 'out'
print("Lines:", lines)
print("Words:", words)
print("Letters:", letters)

```

תוכן של הקובץ:

text1.txt - Notepad

```

File Edit Format View Help
apple banana potato apple
pineapple orange lemon
kiwi lemon orange
potato orange apple

```

תרגיל מס' 6.

תפענח את הקוד. מהו פלט של התוכנית בהתאם לתוכן הקובץ?

```

products = {}
for i in open("C:\\MyPython\\goods.txt"):
    row = i.split()
    row[1] = float(row[1])
    row[2] = int(row[2])
    products[row[0]] = row[1:]
print(products)

```

תוכן של הקובץ:

goods.txt - Notepad

```

File Edit Format View Help
Apple 4.5 10
Orange 6.2 5
Pineapple 10.0 1
Mango 7.5 2
Banana 3.8 10

```

תרגיל מס' 7.

תפענח את הקוד. מהו יהיה תוכן הקובץ לאחר הרת התוכנית?


```
goods = {"Apple": 4.5,
         "Orange": 6.2,
         "Pineapple": 10.0,
         "Mango": 7.5,
         "Banana": 3.8}
f = open("C:\\MyPython\\write_goods.txt", 'w')
template = "{} {} \n"
for fruit, price in goods.items():
    line = template.format(fruit, price)
    f.write(line)
f.close()
```

תרגיל מס' 8.

תפענח את הקוד. מהו יהיה תוכן הקובץ לאחר הרת התוכנית?

```
tab_file = open("C:\\MyPython\\tab.txt")
tab_text = tab_file.read()
tab_file.close()
print(repr(tab_text))
list_text = tab_text.split('\t')
space_text = ' '.join(list_text)
print(repr(space_text))
space_file = open("C:\\MyPython\\space.txt", 'w')
space_file.write(space_text)
space_file.close()
```

קובץ מקורי הוא:

 \*tab.txt - Notepad

File Edit Format View Help

```
if a > b:
    print(c)
while c > 0:
    n = c % 10
    if n > 5:
        print(n)
    c %= 10
```

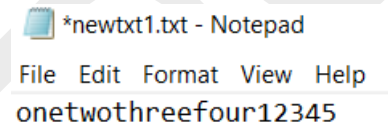


תרגיל מס' 9.

תפענח את הקוד. מהו פלט של התוכנית?

```
def file_read(fname):  
    txt = open(fname)  
    print(txt.read())  
file_read('C:\\MyPython\\newtxt1.txt')
```

תוכן הקובץ הוא:



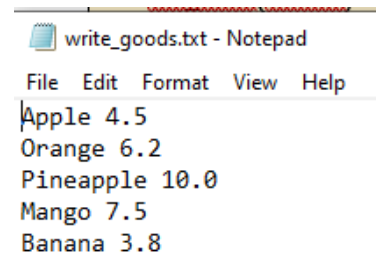
```
*newtxt1.txt - Notepad  
File Edit Format View Help  
onetwothreefour12345
```

תרגיל מס' 10.

תפענח את הקוד. מהו פלט של התוכנית?

```
def file_read_from_head(fname, nlines):  
    from itertools import islice  
    with open(fname) as f:  
        for line in islice(f, nlines):  
            print(line, end="")  
file_read_from_head('C:\\MyPython\\write_goods.txt', 2)
```

תוכן הקובץ הוא:



```
write_goods.txt - Notepad  
File Edit Format View Help  
Apple 4.5  
Orange 6.2  
Pineapple 10.0  
Mango 7.5  
Banana 3.8
```

תרגיל מס' 11.

תפענח את הקוד. מהו פלט של התוכנית?

```
def file_read(fname):  
    from itertools import islice  
    with open(fname, "w") as myfile:  
        myfile.write("Python Files Exercises \n")  
        myfile.write("Python Files Answers ")  
    txt = open(fname)  
    print(txt.read())  
file_read('C:\\MyPython\\abc.txt')
```

תוכן הקובץ הוא:

## תרגיל מס' 12.

תפענח את הקוד. מהו פלט של התוכנית?

```
import sys
import os
def file_read_from_tail(fname,lines):
    bufsize = 8192
    fsize = os.stat(fname).st_size
    iter = 0
    with open(fname) as f:
        if bufsize > fsize:
            bufsize = fsize-1
            data = []
            while True:
                iter +=1
                f.seek(fsize-bufsize*iter)
                data.extend(f.readlines())
                if len(data) >= lines or f.tell() == 0:
                    print(".join(data[-lines:]))
                    break
file_read_from_tail('C:\\MyPython\\write_goods.txt',2)
```

תוכן הקובץ הוא:

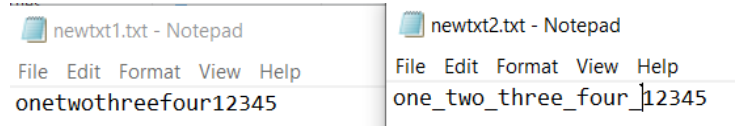
```
write_goods.txt - Notepad
File Edit Format View Help
Apple 4.5
Orange 6.2
Pineapple 10.0
Mango 7.5
Banana 3.8
```

## תרגיל מס' 13.

תפענח את הקוד. מהו פלט של התוכנית?

```
f1=open("C:\\MyPython\\newtxt1.txt","r")
f2=open("C:\\MyPython\\newtxt2.txt","r")
for line1 in f1:
    for line2 in f2:
        if line1==line2:
            print("SAME\n")
        else:
            print(line1 + line2)
```

תוכן הקבצים הם:

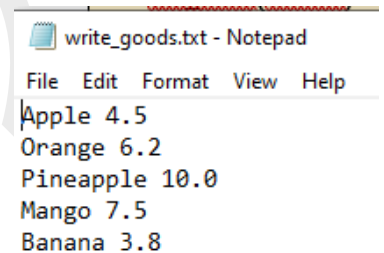


תרגיל מס' 14.

תפענח את הקוד. מהו פלט של התוכנית?

```
def file_read(fname):  
    with open(fname) as f:  
        content_list = f.readlines()  
        print(content_list)  
file_read('C:\\MyPython\\write_goods.txt')
```

תוכן הקובץ הוא:

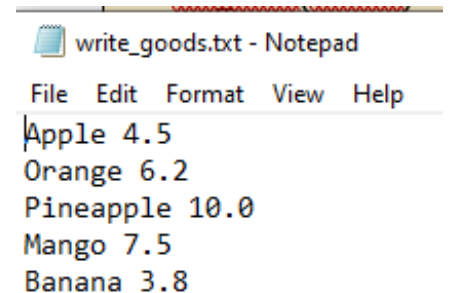


תרגיל מס' 15.

תפענח את הקוד. מהו פלט של התוכנית?

```
filepath = 'C:\\MyPython\\write_goods.txt'  
with open(filepath) as fp:  
    line = fp.readline()  
    cnt = 0  
    while line:  
        print("Line {}: {}".format(cnt, line.strip()))  
        line = fp.readline()  
        cnt += 1
```

תוכן הקובץ הוא:

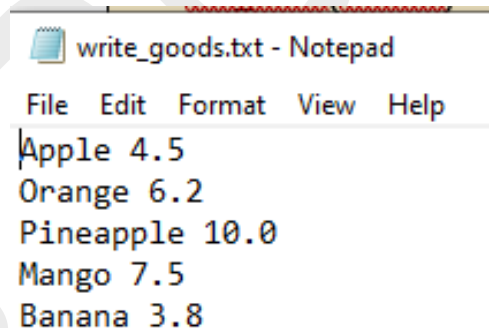


תרגיל מס' 16.

תפענח את הקוד. מהו פלט של התוכנית?

```
def file_read(fname):
    content_array = []
    with open(fname) as f:
        for line in f:
            content_array.append(line)
    print(content_array)
file_read('C:\\MyPython\\write_goods.txt')
```

תוכן הקובץ הוא:



write\_goods.txt - Notepad

File Edit Format View Help

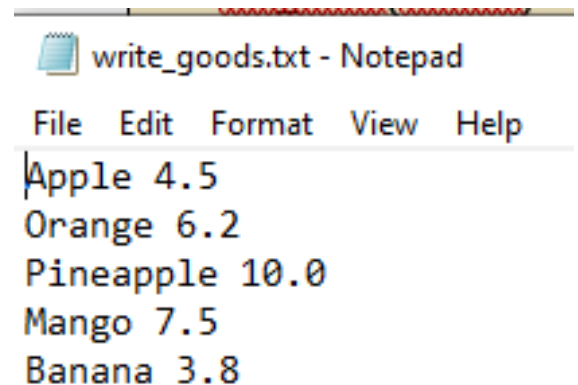
Apple 4.5  
Orange 6.2  
Pineapple 10.0  
Mango 7.5  
Banana 3.8

תרגיל מס' 17.

תפענח את הקוד. מהו פלט של התוכנית?

```
def longest_word(filename):
    with open(filename, 'r') as infile:
        words = infile.read().split()
    max_len = len(max(words, key=len))
    return [word for word in words if len(word) == max_len]
print(longest_word('C:\\MyPython\\write_goods.txt'))
```

תוכן הקובץ הוא:



write\_goods.txt - Notepad

File Edit Format View Help

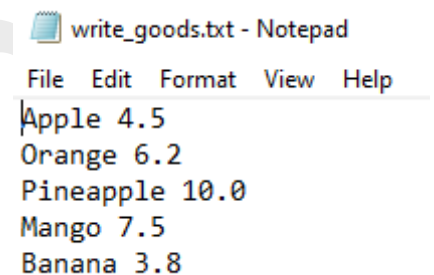
Apple 4.5  
Orange 6.2  
Pineapple 10.0  
Mango 7.5  
Banana 3.8

תרגיל מס' 18.

תפענח את הקוד. מהו פלט של התוכנית?

```
def file_lengthy(fname):  
    with open(fname) as f:  
        for i, l in enumerate(f):  
            pass  
        return i + 1  
print("Number of lines in the file: ", file_lengthy("C:\\MyPython\\write_goods.txt"))
```

תוכן הקובץ הוא:



write\_goods.txt - Notepad

File Edit Format View Help

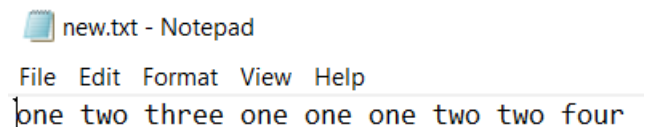
Apple 4.5  
Orange 6.2  
Pineapple 10.0  
Mango 7.5  
Banana 3.8

תרגיל מס' 19.

תפענח את הקוד. מהו פלט של התוכנית?

```
from collections import Counter  
def word_count(fname):  
    with open(fname) as f:  
        return Counter(f.read().split())  
print("Number of words in the file :", word_count("C:\\MyPython\\new.txt"))
```

תוכן הקובץ הוא:



new.txt - Notepad

File Edit Format View Help

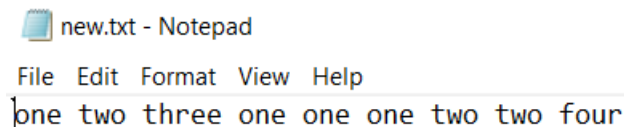
one two three one one one two two four

תרגיל מס' 20.

תפענח את הקוד. מהו פלט של התוכנית?

```
def file_size(fname):  
    import os  
    statinfo = os.stat(fname)  
    return statinfo.st_size  
print("File size in bytes of a plain file: ", file_size("C:\\MyPython\\new.txt"))
```

תוכן הקובץ הוא:



תרגיל מס' 21.

תפענח את הקוד. מהו פלט של התוכנית? מה יהיה תוכן הקובץ abc.txt.

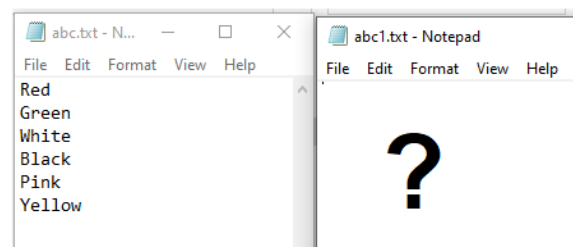
```
color = ['Red', 'Green', 'White', 'Black', 'Pink', 'Yellow']
with open('abc.txt', "w") as myfile:
    for c in color:
        myfile.write("%s\n" % c)
content = open("C:\\MyPython\\abc.txt")
print(content.read())
```

תרגיל מס' 22.

תפענח את הקוד. מהו תוכן של קובץ abc1.txt?

```
f = open("C:\\MyPython\\abc.txt")
f1 = open("C:\\MyPython\\abc1.txt", 'w')
for line in f.readlines():
    f1.write(line)
f1.close()
f.close()
```

תוכן הקובץ הוא:



תרגיל מס' 23.

תפענח את הקוד. תן דוגמה לפלט בהתאם לקלט שתבחר.

```
import random
def random_line(fname):
    lines = open(fname).read().splitlines()
    return random.choice(lines)
print(random_line("C:\\MyPython\\abc1.txt"))
```

תוכן הקובץ הוא:

```
abc1.txt - Notepad
File Edit Format View Help
Red
Green
White
Black
Pink
Yellow
```

תרגיל מס' 24.

תפענח את הקוד. מהו פלט של התוכנית?

```
f = open("C:\\MyPython\\abc1.txt",'r')
print(f.closed)
f.close()
print(f.closed)
```

תרגיל מס' 25.

תפענח את הקוד. מהו פלט של התוכנית?

```
def remove_newlines(fname):
    flist = open(fname).readlines()
    return [s.rstrip("\n") for s in flist]
print(remove_newlines("C:\\MyPython\\abc1.txt"))
```

תוכן הקובץ הוא:

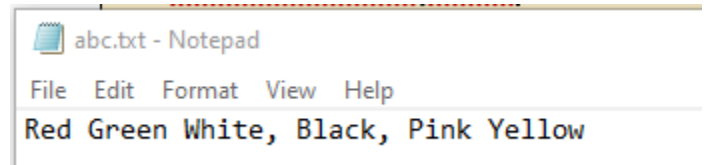
```
abc1.txt - Notepad
File Edit Format View Help
Red
Green
White
Black
Pink
Yellow
```

תרגיל מס' 26.

תפענח את הקוד. מהו פלט של התוכנית?

```
def count_words(filepath):
    with open(filepath) as f:
        data = f.read()
        data.replace(", ", " ")
    return len(data.split(" "))
print(count_words("C:\\MyPython\\abc.txt"))
```

תוכן הקובץ הוא:



תרגיל מס' 27.

תפענח את הקוד. מה עושה תוכנית?

```
import glob
char_list = []
files_list = glob.glob("C:\\MyPython\\*.txt")
for file_elem in files_list:
    with open(file_elem, "r") as f:
        char_list.append(f.read())
print(char_list)
```

תרגיל מס' 28.

תפענח את הקוד. מהו פלט של התוכנית?

```
import string, os
if not os.path.exists("letters"):
    os.makedirs("letters")
for letter in string.ascii_uppercase:
    with open("C:\\MyPython\\"+ letter + ".txt", "w") as f:
        f.writelines(letter)
```

תרגיל מס' 29.

תפענח את הקוד. מהו פלט של התוכנית?

```
import string
def letters_file_line(n):
    with open("C:\\MyPython\\words1.txt", "w") as f:
        alphabet = string.ascii_uppercase
        letters = [alphabet[i:i + n] + "\n" for i in range(0, len(alphabet), n)]
        f.writelines(letters)
letters_file_line(3)
```



## תרגול בנושא כתיבת קוד.

תרגיל מס' 1.

כתוב תוכנית שקוראת קובץ טקסט ומעתיקה את התוכן לרשימה של מחרוזות. יש להכין מראש קובץ עם תוכן – להכניס מספר שורות של טקסט, לשמור אותו ולשחרר אותו (יש לסגור את Notepad לפני).

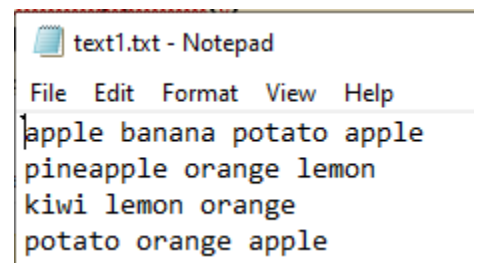
לדוגמה:

```
['it is funny\n', 'to program python\n', 'but it is also funny \n', 'to program C']  
['it is funny', 'to program python', 'but it is also funny ', 'to program C']
```

תחזור על תהליך ותמחק סימן '\n' בעזרת שיטת `.strip()`.

תרגיל מס' 2.

כתוב תוכנית שבודקת כמה פעמים מופיע מילה בתוך קובץ טקסט. הכנתי קובץ עם תוכן הבא:



```
text1.txt - Notepad  
File Edit Format View Help  
apple banana potato apple  
pineapple orange lemon  
kiwi lemon orange  
potato orange apple
```

(ניתן להוציא נתונים מהקובץ בלי להסתבך, בעזרת שיטת `.read()`. לקבצים גדולים זה לא מתאים, כי זה יכול למלא את אזור הזיכרון מהר מאוד.)

שימו לב, שאנו סופרים מילים ולא מספר תת-מחרוזות בתוך מחרוזות. אם הינו צריכים למצוא תת-מחרוזות הינו משתמשים בשיטת `find()` שמחזירה מיקום של אובייקט לפי תו הראשון של תת-מחרוזות.

תרגיל מס' 3.

כתוב תוכנית שבודקת כמה מחרוזות, כמה מילים וכמה תווים יש בתוך הקובץ טקסט.

פותחים את הקובץ וקוראים את הקובץ הפתוח שורה אחרי שורה. אם קוראים את השורה הבאה, אז מונה השורות גודל. אורך השורה, כלומר מספר התווים בו, נקבע באמצעות הפונקציה המובנית `.len()`.

כדי לספור את מספר המילים, מוצג משתנה - דגל, בעזרתו נקבע האם התו הבא נמצא בתוך מילה או שזו תחילתה של מילה חדשה. אם התו אינו רווח והדגל מציין שהוא נמצא מחוץ למילה, אז ספירת המילים מצטברת והדגל מוגדר "בתוך מילה". בכל פעם שמתרחש רווח, הדגל מתאפס למצב "מחוץ למילה".

#### תרגיל מס' 4.

נתון קובץ טקסט שכל שורה מכילה את שם המוצר, מחירו וכמותו. כתוב תוכנית שמכניסה את תוכן הקובץ למילון כך שכל רשומת מילון תואמת שורה אחת בקובץ. מפתח הרשימה הוא שם המוצר, הערך הוא הרשימה, שהאיבר הראשון שלו הוא המחיר, השני הוא כמות המוצר.

מכינים קובץ goods.txt בצורה הבאה:

```
goods.txt - Notepad
File Edit Format View Help
Apple 4.5 10
Orange 6.2 5
Pineapple 10.0 1
Mango 7.5 2
Banana 3.8 10
```

הפלט צריך להיות:

```
{'Apple': [4.5, 10], 'Orange': [6.2, 5], 'Pineapple': [10.0, 1], 'Mango': [7.5, 2], 'Banana': [3.8, 10]}
```

#### תרגיל מס' 5.

מוגדר מילון שהמפתחות שלו הם סחורה, והערכים הם המחירים שלהם. כתוב את נתוני המילון לקובץ כך שכל זוג איברים של מילון יהיה בשורה נפרדת.

#### תרגיל מס' 6.

בקובץ טקסט יש להחליף סימני TAB ב-4 רווחים.

נכין קובץ tab.txt:

```
*tab.txt - Notepad
File Edit Format View Help
if a > b:
    print(c)
while c > 0:
    n = c % 10
    if n > 5:
        print(n)
    c %= 10
```

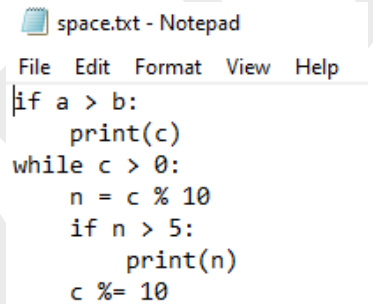
#### תרגיל מס' 7.

פותחים את הקובץ וקולטים את כל התוכן של הקובץ בעזרת שיטת read(). מעתיקים את התוכן של הקובץ לתוך משתנה עזר וסוגרים את הקובץ. בעזרת פונקציה repr() מדפיסים את כל הסימנים שאי-אפשר לראות. טקסט מועבר לרשימה במקום שנמצא סימן "t".

לאחר סיום התהליך מאחדים את הרשימה לטקסט. בין איברים של טקסט מכניסים 4 סימני רווח במקום סימן '\'.  
הפלט של התוכנית בתחילת התוכנית ובסוף התוכנית צריך להיות:

```
'if a > b:\n\tprint(c)\nwhile c > 0:\n\tn = c % 10\n\tif n > 5:\n\t\tprint(n)\n\tc %= 10'  
'if a > b:\n print(c)\nwhile c > 0:\n n = c % 10\n if n > 5:\n print(n)\n c %= 10'
```

קובץ מקורי הוא:



```
space.txt - Notepad  
File Edit Format View Help  
if a > b:  
    print(c)  
while c > 0:  
    n = c % 10  
    if n > 5:  
        print(n)  
    c %= 10
```

תרגיל מס' 8.

כתוב תוכנית שקוראת קובץ טקסט בעזרת פונקציה שמקבלת שם הקובץ ומדפיסה את תוכן הקובץ.

תרגיל מס' 9.

כתוב תוכנית לקריאת n שורות ראשונות של הקובץ בעזרת פונקציה שמקבלת שם הקובץ וכמות שורות להדפסה ומדפיסה את התוצאה.

תרגיל מס' 10.

כתוב תוכנית שמוסיפה טקסט לקובץ ומציגה אותו על המסך.

תרגיל מס' 11.

כתוב תוכנית לקריאת n שורות אחרונות של הקובץ.

תרגיל מס' 12.

כתוב תוכנית שמשווה תוכן של שני קבצים ומחזירה תשובה האם הם שווים או לא. אם לא, תוכנית מדפיסה תוכן של שני קבצים ביחד.

תרגיל מס' 13.

כתוב תוכנית שקוראת קובץ שורה אחרי שורה ומכניסה את התוכן לרשימה.

לדוגמה:

```
['Apple 4.5\n', 'Orange 6.2\n', 'Pineapple 10.0\n', 'Mango 7.5\n', 'Banana 3.8\n']
```

תרגיל מס' 14.

כתוב תוכנית שקוראת קובץ שורה אחרי שורה ומכניסה את התוכן למשתנה.  
לדוגמה:

```
Line 0: Apple 4.5  
Line 1: Orange 6.2  
Line 2: Pineapple 10.0  
Line 3: Mango 7.5  
Line 4: Banana 3.8
```

תרגיל מס' 15.

כתוב תוכנית שקוראת קובץ שורה אחרי שורה ומכניסה את התוכן למערך.  
לדוגמה:

```
['Apple 4.5\n', 'Orange 6.2\n', 'Pineapple 10.0\n', 'Mango 7.5\n', 'Banana 3.8\n']
```

תרגיל מס' 16.

כתוב תוכנית למציאת מילה הארוכה ביותר בקובץ טקסט.

תרגיל מס' 17.

כתוב תוכנית לספירת שורות של קובץ טקסט.

```
Number of lines in the file: 5
```

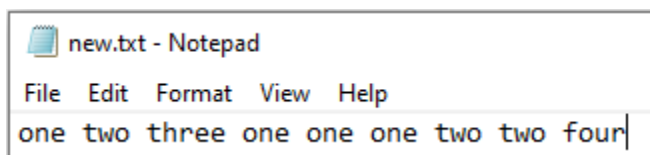
תרגיל מס' 17.

כתוב תוכנית שסופרת את תדירות 3 מילים בתוך קובץ טקסט.

לדוגמה פלט:

```
Number of words in the file : Counter({'one': 4, 'two': 3, 'three': 1, 'four': 1})
```

עבור קובץ:



```
new.txt - Notepad  
File Edit Format View Help  
one two three one one one two two four
```

תרגיל מס' 19.

כתוב תוכנית לחישוב גודל הקובץ.

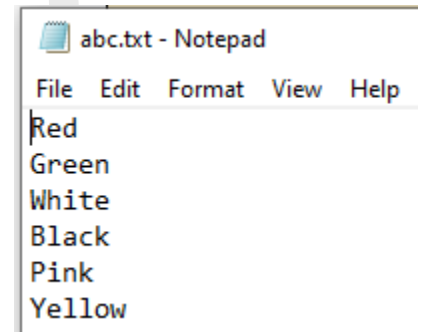
תרגיל מס' 20.

כתוב תוכנית לכתיבת רשימה לקובץ.

פלט לדוגמה:

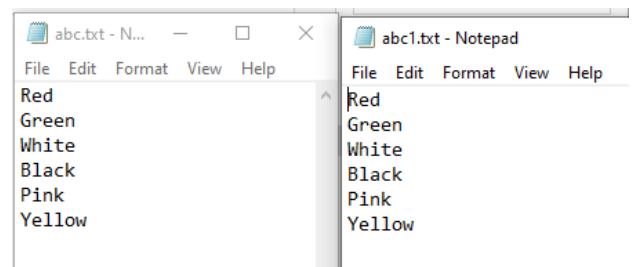
```
Red
Green
White
Black
Pink
Yellow
```

תוכן הקובץ:



תרגיל מס' 21.

כתוב תוכנית שמעתיקה את תוכן הקובץ לתוך קובץ אחר.



תרגיל מס' 22.

כתוב תוכנית שמאחדת שורה של קובץ ראשון עם שורה של קובץ שני וכך עד סוף הקובץ.

לדוגמה:

```
Red
Red
Green
```

Green  
White  
White  
Black  
Black  
Pink  
Pink  
Yellow  
Yellow

תרגיל מס' 23.

כתוב תוכנית שקוראת שורה אקראית של הקובץ.

תרגיל מס' 24.

כתוב תוכנית שבודקת האם הקובץ סגור או לא.

תרגיל מס' 25.

כתוב תוכנית שמורידה סימני "\n" מהקובץ.

לדוגמה:

```
['Red', 'Green', 'White', 'Black', 'Pink', 'Yellow']
```

תרגיל מס' 26.

כתוב תוכנית שמקבלת את שם הקובץ ומחזירה את כמות המילים שיש בקובץ. שימו לב שכמה מילים יכולים להיות נפרדות בעזרת פסיקים ולא עם רווחים.

תרגיל מס' 27.

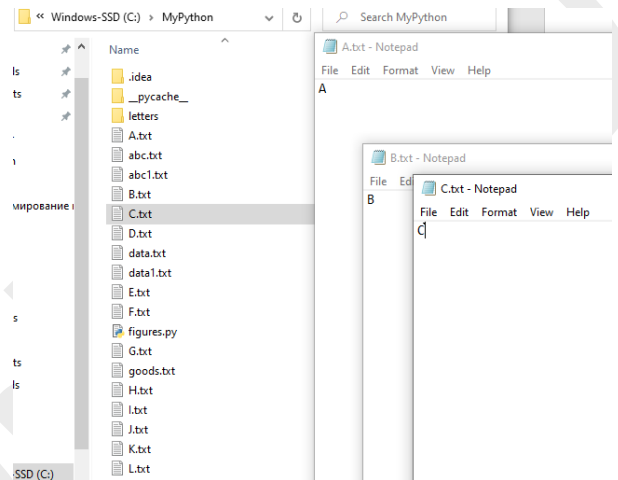
כתוב תוכנית שלוקחת תווים מהקבצים שנמצאים בספרייה ומעתיקה אותם לרשימה.

לדוגמה:

```
['Red Green White, Black, Pink Yellow\n', 'Red\nGreen\nWhite\nBlack\nPink\nYellow\n', '4 rrrrrr',  
'50; 30; 40; 35; ', 'Apple 4.5 10\nOrange 6.2 5\nPineapple 10.0 1\nMango 7.5 2\nBanana 3.8  
10', 'one two three one one one two two four', 'onetwothreefour12345', 'onetwothreefour12345',  
'Pretty cool, huh!\n', 'if a > b:\n print(c)\nwhile c > 0:\n n = c % 10\n if n > 5:\n print(n)\n c %= 10', 'if a > b:\n\tprint(c)\nwhile c > 0:\n\t n = c % 10\n\tif n > 5:\n\t\t\tprint(n)\n\t\t c %= 10', 'it is funny\nto program python\nbut it is also funny \nto program C', 'apple banana potato  
apple\npineapple orange lemon\nkiwi lemon orange\npotato orange apple', 'Apple 4.5\nOrange  
6.2\nPineapple 10.0\nMango 7.5\nBanana 3.8\n']
```

תרגיל מס' 28.

כתוב תוכנית שמייצרת 26 קבצי טקסט עם שמות: A.txt, B.txt, C.txt וכו' עד Z.txt. בכל קובץ מצאת את זהה לשם הקובץ.



תרגיל מס' 29.

כתוב תוכנית ליצירת קובץ, שמכיל את כל האותיות של שפה אנגלית לפי סדר א'-ב'.

תרגיל מס' 30.

תייצר קובץ data.txt ותכתוב לתוכו מילים באנגלית. כתוב תוכנית שפותחת קובץ במצב קריאה, קוראת ממנו נתונים וכותבת אותם על המסך. תוכנית קולטת מהמשתמש תרגום לעברית ומעתיק אותם לקובץ אחר חדש יחד עם מילים של מקור בזוגות מסוג צירוף.

תרגיל מס' 31.

תכין קובץ nums.txt שמכיל מספרים עם רווחים ביניהם. כתוב תוכנית שקוראת מספרים האלה, מחשבת סכומם, מדפיסה את הסכום ושומרת את הסכום בסוף הקובץ.

## בונוס:

תרגיל מס' 1.

תוכנית לערבוב חפיסת קלפים:

```
# Python program to shuffle a deck of card
# importing modules
import itertools, random
# make a deck of cards
deck = list(itertools.product(range(1,14),['Spade','Heart','Diamond','Club']))
# shuffle the cards
random.shuffle(deck)
# draw five cards
print("You got:")
for i in range(5):
    print(deck[i][0], "of", deck[i][1])
```

```
You got:
8 of Club
9 of Heart
3 of Heart
12 of Spade
10 of Spade
```

תרגיל מס' 2. BlackJack פשוט לשחקן יחיד – משחק אחד.

```
cards = [6,7,8,9,10,2,3,4,11] * 4
import random
random.shuffle(cards)
print('Lets play BlackJack?')
count = 0
while True:
    choice = input('Do You want to take the card? y/n\n')
    if choice == 'y':
        current = cards.pop()
        print('You have the %d card ' %current)
        count += current
        if count > 21:
            print('Sorry You lost the game')
            break
        elif count == 21:
            print('Great You have 21!')
            break
        else:
            print('You have %d point.' %count)
    elif choice == 'n':
```



```
print('You finished the game with %d points.' %count)
break

print('Have a nice day!')
```

```
Lets play BlackJack?
Do You want to take the card? y/n
y
You have the 3 card
You have 3 point.
Do You want to take the card? y/n
y
You have the 9 card
You have 12 point.
Do You want to take the card? y/n
y
You have the 8 card
You have 20 point.
Do You want to take the card? y/n
n
You finished the game with 20 points.
Have a nice day!
```

תרגיל מס' 3.

כתוב המשך של תוכנית מס' 1 שמדמיית משחק אחד בין מחשב לבין שחקן. משחק מדפיס כל מהלך של מחשב ושל שחקן ובסוף מדפיס תוצאה מי ניצח.

תרגיל מס' 4.

כתוב המשך של תוכנית מס' 2 שמדמיית משחק בין מחשב לבין שחקן. בתחילת המשחק שואלים את שחקן, איזה סכום יש לו, כל פעם שואלים אותו כמה כסף הוא רוצה לשים. משחק ממשיך עד שלשחקן יש כסף או עד שהשחקן רוצה להמשיך. בסוף המשחק מדפיסים כמה כסף הפסיד/הרוויח השחקן וכמה משחקים זכה השחקן ובכמה זכה המחשב.

תרגיל מס' 5.

ציור של יין-יאנג בשפת פייתון:

```
from turtle import *

RAD = 100
RAD2 = RAD / 2
RAD6 = RAD / 6

degrees() # Switch to degrees
# Draw the circle, radius 100, half black
fillcolor('black')
```

```

begin_fill()
circle(RAD, 180)
end_fill()
circle(RAD, 180)

# Draw smaller black semi-circle
left(180)
penup()
goto(0, RAD)
pendown()
begin_fill()
circle(RAD2, 180)
end_fill()

# Draw smaller white semi-circle
penup()
goto(0, RAD)
pendown()
fillcolor('white')
begin_fill()
circle(RAD2, 180)
end_fill()

# Draw smaller circles
penup()
goto(0, RAD2 + RAD6)
begin_fill()
circle(RAD6)
end_fill()

fillcolor('black')
goto(0, 2 * (RAD - RAD6))
begin_fill()
circle(RAD6)
end_fill()

```



```
import random
flag=1
winC=0
winU=0
left=1
right=100
user_num=int(input("Please enter Your number from 1 to 100"))
comp_num=random.randrange(1,100,1)
#print(comp_num)
while flag>0:
    print("Please enter the number from 1 to 100")
    player_num=int(input())
    if player_num>comp_num:
        print("Your number is greater")
    elif player_num<comp_num:
        print("Your number is smaller")
    else:
        print("Bingo")
        winU=1
        flag=0
    answ=int((left+right)/2)
    print("My answ is ", answ)
    if answ>user_num:
        right=answ
    elif answ<user_num:
        left=answ
    else:
        print("Bingo")
        winC=1
        flag=0
if winU*winC==1:
    print("Teko ")
elif winC==1:
    print("Comp winner")
else:
    print("User winner")
```

תנסו לבדוק את עצמכם ולענות על השאלון הקצר:

שאלון בנושא מבוא ומשתנים:

<https://docs.google.com/forms/d/1EWiBxMo8TY8qXQyZnmkutVn0gJM09G-JIyjizXebF0I>

שאלון בנושא סוגי נתונים:

[https://docs.google.com/forms/d/1fWETxg0WBvTdzGuHNL\\_CmXBt5NNB3i6-bpqqRmTFjno](https://docs.google.com/forms/d/1fWETxg0WBvTdzGuHNL_CmXBt5NNB3i6-bpqqRmTFjno)

שאלון בנושא קבצים:

<https://docs.google.com/forms/d/1EsuPNTjCSJcMjPGyFR0rZ2DKzZoVaxBZZCfGiKGomt0/prefill>

שאלון בנושא מחרוזות:

[https://docs.google.com/forms/d/1Vo-452cQ\\_5l3eR7-uEBAq\\_xhe7i3syT\\_o6y0Und--mI/prefill](https://docs.google.com/forms/d/1Vo-452cQ_5l3eR7-uEBAq_xhe7i3syT_o6y0Und--mI/prefill)

שאלון בנושא סטים:

<https://docs.google.com/forms/d/1L1ID7Xg-clCmkRvOcQY5T28GhgFgAkoz5yPHPncJ7uA/prefill>

©2012-2015 - Laurent Pointal Memento v2.0.6 License Creative Commons Attribution 4

## Python 3 Cheat Sheet

Latest version on : <https://perso.linux.fr/pointal/python-memento>

### Base Types

integer, float, boolean, string, bytes

```
int 783 0 -192 0b010 0o642 0xF3
float 9.23 0.0 -1.7e-6
bool True False
str "One\nTwo"
bytes b"toto\xfe\xff"
```

*hexadecimal octal binary decimal hexa*

*escaped new line*

*escaped*

*hexadecimal octal*

*immutables*

### Container Types

- ordered sequences**, fast index access, repeatable values
  - list** [1, 5, 9]
  - tuple** (1, 5, 9)
  - str bytes** ("x", 11, 8.9)
- key containers**, no a priori order, fast key access, each key is unique
  - dictionary dict** {"key": "value"}
  - set** {"key1", "key2"}
  - frozenset** immutable set

*Non modifiable values (immuables)*

*expression with only commas → tuple*

*(key/value associations)*

*keys=hashable values (base types, immuables...)*

*empty*

### Identifiers

for variables, functions, modules, classes... names

- a-zA\_Z\_** followed by **a-zA\_Z\_0-9**
- diacritics allowed but should be avoided
- language keywords forbidden
- lower/UPPER case discrimination
- a toto x7 y\_max BigOne**
- By and Go**

### Conversions

**type (expression)**

- int** ("15") → 15
- int** ("3f", 16) → 63
- int** (15.56) → 15
- float** ("-11.24e8") → -1124000000.0
- round** (15.56, 1) → 15.6
- bool** (x) → False for null x, empty container x, None or False x; True for other x
- str** (x) → "..." representation string of x for display (cf. formatting on the back)
- chr** (64) → '@' **ord** ('@') → 64
- repr** (x) → "..." literal representation string of x
- bytes** ([72, 9, 64]) → b'H\t@'
- list** ("abc") → ['a', 'b', 'c']
- dict** ({3, "three"}, (1, "one")) → {1: 'one', 3: 'three'}
- set** ({ "one", "two" }) → {'one', 'two'}
- separator str** and **sequence of str** → **assembled str**
- str** splitted on whitespaces → **list of str**
- "words with spaces".split()** → ['words', 'with', 'spaces']
- str** splitted on **separator str** → **list of str**
- "1,4,8,2".split(",")** → ['1', '4', '8', '2']
- sequence of one type** → **list of another type (via list comprehension)**
- [int(x) for x in ('1', '29', '-3')]** → [1, 29, -3]

### Variables assignment

=

! assignment ↔ **binding** of a name with a value

- evaluation of right side expression value
- assignment in order with left side names

**x=1.2+8+sin(y)**

**a=b=c=0** assignment to same value

**y, z, r=9.2, -7.6, 0** multiple assignments

**a, b=b, a** values swap

**a, \*b=seq** unpacking of sequence in item and list

**\*a, b=seq** unpacking of sequence in item and list

**x+=3** increment ↔ **x=x+3** and

**x-=2** decrement ↔ **x=x-2** /=

**x=None** undefined = constant value %

**del x** remove name x ...

### Sequence Containers Indexing

for lists, tuples, strings, bytes...

negative index	-5	-4	-3	-2	-1
positive index	0	1	2	3	4

**lst=[10, 20, 30, 40, 50]**

positive slice: 0 1 2 3 4 5

negative slice: -5 -4 -3 -2 -1

**Items count** **len(lst) → 5**

! index from 0 (here from 0 to 4)

Individual access to **items** via **lst [index]**

**lst[0] → 10** → first one **lst[1] → 20**

**lst[-1] → 50** → last one **lst[-2] → 40**

On mutable sequences (**list**), remove with **del lst[3]** and modify with assignment **lst[4]=25**

Access to **sub-sequences** via **lst [start slice : end slice : step]**

**lst[-1:] → [10, 20, 30, 40]** **lst[: -1] → [50, 40, 30, 20, 10]** **lst[1:3] → [20, 30]** **lst[:3] → [10, 20, 30]**

**lst[1: -1] → [20, 30, 40]** **lst[: -2] → [50, 30, 10]** **lst[-3: -1] → [30, 40]** **lst[3:] → [40, 50]**

**lst[: : 2] → [10, 30, 50]** **lst[:] → [10, 20, 30, 40, 50]** shallow copy of sequence

Mixing slice indication → from start / up to end.

On mutable sequences (**list**), remove with **del lst[3:5]** and modify with assignment **lst[1:4]=[15, 25]**

### Boolean Logic

Comparisons: < > <= >= == != (boolean results)

**a and b** logical and both simultaneously

**a or b** logical or one or other or both

! pitfall : **and** and **or** return value of a or of b (under shortest evaluation).

⇒ ensure that **a** and **b** are booleans.

**not a** logical not

**True** **False** } True and False constants

### Statements Blocks

```
parent statement:
statement block 1...
...
parent statement:
statement block2...
next statement after block 1
```

! configure editor to insert 4 spaces in place of an indentation tab.

### Modules/Names Imports

**module true** ↔ file **true.py**

**from monmod import nom1, nom2 as fct** → direct access to names, renaming with **as**

**import monmod** → access via **monmod.nom1...**

! modules and packages searched in python path (cf **sys.path**)

### Conditional Statement

statement block executed only if a condition is true

**if logical condition:** statements block

Can go with several **elif, elif...** and only one final **else**. Only the block of first true condition is executed.

! with a var **x**:

**if bool(x)==True:** ↔ **if x:**

**if bool(x)==False:** ↔ **if not x:**

```
if age <= 18:
    state = "Kid"
elif age > 65:
    state = "Retired"
else:
    state = "Active"
```

### Maths

angles in radians

```
from math import sin, pi
sin(pi/4) → 0.707...
cos(2*pi/3) → -0.4999...
sqrt(81) → 9.0
log(e**2) → 2.0
ceil(12.5) → 13
floor(12.5) → 12
```

modules **math, statistics, random**, decimal, fractions, numpy, etc. (cf. doc)

### Exceptions on Errors

Signaling an error: **raise ExcClass(...)**

Errors processing:

```
try:
    normal processing block
except Exception as e:
    error processing block
```

! finally block for final processing in all cases.

### Conditional Loop Statement

statements block executed as long as condition is true

```
while logical condition:
    statements block
```

*be aware of infinite loops!*

```
s = 0
i = 1
while i <= 100:
    s = s + i**2
    i = i + 1
print("sum:", s)
```

initializations before the loop condition with a least one variable value (here i)

make condition variable change!

Algo:  $s = \sum_{i=1}^{100} i^2$

### Iterative Loop Statement

statements block executed for each item of a container or iterator

```
for var in sequence:
    statements block
```

Go over sequence's values

```
s = "Some text"
cnt = 0
for c in s:
    if c == "e":
        cnt = cnt + 1
print("found", cnt, "e")
```

initializations before the loop loop variable, assignment managed by for statement

Algo: count number of e in the string.

### Display

```
print("v=", 3, "cm :", x, ", ", y+4)
```

items to display: literal values, variables, expressions

print options:

- sep=" " items separator, default space
- end="\n" end of print, default new line
- file=sys.stdout print to file, default standard output

### Input

```
s = input("Instructions: ")
```

input always returns a string, convert it to required type (cf. boxed Conversions on the other side).

loop on dict/set => loop on keys sequences use slices to loop on a subset of a sequence

Go over sequence's index

- modify item at index
- access items around index (before / after)

```
lst = [11, 18, 9, 12, 23, 4, 17]
last = []
for idx in range(len(lst)):
    val = lst[idx]
    if val > 15:
        last.append(val)
        lst[idx] = 15
print("modif:", lst, "-last:", last)
```

Algo: limit values greater than 15, memorizing of last values.

Go simultaneously over sequence's index and values:

```
for idx, val in enumerate(lst):
```

### Generic Operations on Containers

len(c) -> items count  
min(c) max(c) sum(c)  
sorted(c) -> list sorted copy  
val in c -> boolean, membership operator in (absence not in)  
enumerate(c) -> iterator on (index, value)  
zip(c1, c2...) -> iterator on tuples containing c, items at same index  
all(c) -> True if all c items evaluated to true, else False  
any(c) -> True if at least one item of c evaluated true, else False

Specific to ordered sequences containers (lists, tuples, strings, bytes...)

```
reversed(c) -> reversed iterator c*5 -> duplicate c+c2 -> concatenate
c.index(val) -> position c.count(val) -> events count
```

import copy  
copy.copy(c) -> shallow copy of container  
copy.deepcopy(c) -> deep copy of container

Note: For dictionaries and sets, these operations use keys.

### Integer Sequences

```
range([start,] end [,step])
```

start default 0, end not included in sequence, step signed, default 1

```
range(5) -> 0 1 2 3 4 range(2, 12, 3) -> 2 5 8 11
range(3, 8) -> 3 4 5 6 7 range(20, 5, -5) -> 20 15 10
range(len(seq)) -> sequence of index of values in seq
range provides an immutable sequence of int constructed as needed
```

### Operations on Lists

```
lst.append(val) add item at end
lst.extend(seq) add sequence of items at end
lst.insert(idx, val) insert item at index
lst.remove(val) remove first item with value val
lst.pop([idx]) -> value remove & return item at index idx (default last)
lst.sort() lst.reverse() sort / reverse list in place
```

### Function Definition

```
def fct(x, y, z):
    """documentation"""
    # statements block, res computation, etc.
    return res
```

function name (identifier) named parameters

# statements block, res computation, etc.

return res -> result value of the call, if no computed result to return: return None

parameters and all variables of this block exist only in the block and during the function call (think of a "black box")

Advanced: def fct(x, y, z, \*args, a=3, b=5, \*\*kwargs):

- \*args variable positional arguments (-> tuple), default values.
- \*\*kwargs variable named arguments (-> dict)

### Operations on Dictionaries

```
d[key]=value d.clear()
d[key] -> value del d[key]
d.update(d2) { update/add associations
d.keys() } -> iterable views on
d.values() keys/values/associations
d.items()
d.pop(key[,default]) -> value
d.popitem() -> (key, value)
d.get(key[,default]) -> value
d.setdefault(key[,default]) -> value
```

### Operations on Sets

Operators:

- | -> union (vertical bar char)
- & -> intersection
- ^ -> difference/symmetric diff.
- < <= > >= -> inclusion relations

Operators also exist as methods.

```
s.update(s2) s.copy()
s.add(key) s.remove(key)
s.discard(key) s.clear()
s.pop()
```

### Function Call

```
r = fct(3, i+2, 2*i)
```

storage/use of returned value one argument per parameter

fct is the use of function name with parentheses which does the call

Advanced: \*sequence \*\*dict

### Files

storing data on disk, and reading it back

```
f = open("file.txt", "w", encoding="utf8")
```

file variable for operations	name of file (+path...)	opening mode	encoding of chars for text files:
		'r' read	utf8 ascii
		'w' write	utf8 latin1
		'a' append	

cf. modules os, os.path and pathlib

#### writing

```
f.write("coucou")
f.writelines(list of lines)
```

#### reading

```
f.read([n]) -> next chars
if n not specified, read up to end!
f.readlines([n]) -> list of next lines
f.readline() -> next line
```

text mode t by default (read/write str), possible binary mode b (read/write bytes). Convert from/to required type!

```
f.close() # dont forget to close the file after use!
```

```
f.flush() write cache f.truncate([size]) resize
reading/writing progress sequentially in the file, modifiable with:
f.tell() -> position f.seek(position[,origin])
```

Very common: opening with a guarded block and reading loop on lines of a text file:

```
with open(...) as f:
    for line in f:
        # processing of line
```

### Operations on Strings

```
s.startswith(prefix[,start,end])
s.endswith(suffix[,start,end]) s.strip([chars])
s.count(sub[,start,end]) s.partition(sep) -> (before, sep, after)
s.index(sub[,start,end]) s.find(sub[,start,end])
s.is_*() tests on chars categories (ex. s.isalpha())
s.upper() s.lower() s.title() s.swapcase()
s.casefold() s.capitalize() s.center([width,fill])
s.ljust([width,fill]) s.rjust([width,fill]) s.zfill([width])
s.encode(encoding) s.split([sep]) s.join(seq)
```

### Formatting

```
"modele() {} {}".format(x, y, z) -> str
```

selection: formatting! conversion

```
2
nom
0. nom
4[key]
0[2]
```

Examples:

```
"{:+2.3f}".format(45.72793) -> '+45.728'
"{1:>10s}".format(8, "toto") -> '      toto'
"{x!r}".format(x="I'm") -> "'I'm'"
"'I\''m'"
```

Formatting: fill char alignment sign min width - precision - max width type

<> ^ \_ + - space 0 at start for filling with 0

integer: b binary, c char, d decimal (default), o octal, x or X hexa...  
float: e or E exponential, f or F fixed point, g or G appropriate (default),  
string: s ... % percent

Conversion: s (readable text) or r (literal representation)

a good habit: don't modify loop variable