

## תרגיל בית 3

### הנחיות כלליות:

- קראו **היטב** את השאלות והקפידו שהתוכניות שלכם פועלות בהתאם לנדרש.
- את התרגיל יש לפתור לבד!
- הקפידו על כללי ההגשה המפורסמים באתר. בפרט, יש להגיש את כל השאלות יחד בקובץ `ex3_012345678.py` המצורף לתרגיל, לאחר החלפת הספרות 012345678 במספר ת.ז שלכם, כל 9 הספרות כולל ספרת הביקורת.
- מועד אחרון להגשה: כמפורסם באתר.
- בדיקה עצמית: כדי לוודא את נכונותן ואת עמידותן של התוכניות לקלטים שגויים, בכל שאלה, הריצו את תוכניתכם עם מגוון קלטים שונים, אלה שהופיעו כדוגמאות בתרגיל וקלטים נוספים עליהם חשבתם (וודאו כי הפלט נכון וכי התוכנית אינה קורסת).
- היות ובדיקת התרגילים עשויה להיות אוטומטית, **יש להקפיד על פלטים מדויקים על פי הדוגמאות (עד לרמת הרווח).**
- אופן ביצוע התרגיל: בתרגיל זה עליכם להשלים את הקוד בקובץ המצורף.
- אין למחוק את ההערות שמופיעות בשלד.
- אין לשנות את שמות המשתנים/פונקציות שכבר מופיעים בקובץ השלד של התרגיל.
- **אין להשתמש בספריות חיצוניות (ובפונקציות שלהן).** כלומר, אין להשתמש בפקודת `import`. **כל פונקציה שלא דורשת פקודה זו מותרת לשימוש** (כלומר, זו פונקציה שהמתרגם (interpreter) מכיר ללא פקודה זו).
- **שימו לב כי שלד התרגיל וקבצי דוגמאות של קלט ופלט (קבצי טקסט) נמצאים באותה תיקיה**
- **שימו לב כי במידה ואתם בודקים קובץ קלט עם נתיב מלא בWindows, יש להקפיד על backslash (\) כפול בנתיב הקובץ. לדוגמא, על מנת לגשת לקובץ הנמצא בנתיב `C:\users\me\file.py` יש להעביר את המחרוזת `"C:\\users\\me\\file.py"`**

## שאלה 1-

ממשו את הפונקציה -

count\_words(file\_name)

המקבלת פרמטר יחיד בשם file\_name, המכיל מחרוזת המציינת נתיב של קובץ הקלט המכיל טקסט כלשהו (ניתן להניח כי הנתיב חוקי ומפנה לקובץ קיים). הפונקציה קוראת את קובץ הקלט ויוצרת קובץ טקסט חדש (באותו הנתיב) בשם q1\_output.txt שבו כל שורה מכילה את מספר המילים בשורה המתאימה בטקסט הקלט אשר מכילים אותיות בלבד (כלומר ללא מספרים או תווים מיוחדים כגון "(!-"). מאורך הגדול או שווה 3 אותיות. נזכיר כי לפי הגדרה מילים מופרדות ברווח בלבד, ולא ע"י כל תו מיוחד אחר, ולפי הגדרה שורות בטקסט מופרדות ע"י התו "\n". ניתן להניח כי קובץ הקלט הוא בפורמט התקין. בנוסף לשם הפשטות ניתן להניח כי קובץ האינפוט מכיל אותיות באנגלית בלבד, מספרים ותווים מיוחדים (כלומר לא מכיל אותיות בשפות נוספות).

לדוגמא: מצורף לתרגיל הקובץ Winnie\_the\_Pooh.txt אשר מכיל חלק מספר הילדים פו הדב.

עבור הקובץ הנ"ל, 5 השורות הראשונות של q1\_output.txt יהיו:

1  
0  
10  
9  
9

ו- 5 השורות האחרונות יהיו:

0  
3  
0  
7  
3

הסבר לדוגמא: השורה השלישית בטקסט היא:

If you happen to have read another book about Christopher Robin, you may

כעת אנחנו רוצים לספור מילים שלא מכילות מספרים או תווים מיוחדים והאורך שלהם הוא גדול או שווה ל3 אותיות. לכן נספור את המילים: you, happen, have, read, another, book, about, Christopher, you, may בשורה השלישית של קובץ הפלט נכתוב 10. את המילה Robin, לא החשבנו משום שמכילה תו מיוחד שהוא פסיק (,).

## שאלה 2

א.

ריק משער כי קיימים אינסוף זוגות של ראשוניים אשר ההפרש ביניהם הוא 2. לדוגמא: (3,5), (11,13), (59,61). והחליט לקרוא לכל זוג שכזה "twin pricks"

הוא אפילו כתב על זה מאמר בויקיפדיה :  
[https://en.wikipedia.org/wiki/Twin\\_prime](https://en.wikipedia.org/wiki/Twin_prime)

ממשו את הפונקציה `twin_pricks(num)` אשר כותבת לתוך קובץ בשם `q2a_output_num.txt`, את `num` זוגות התאומים הראשוניים הראשונים בפורמט המתואר בדוגמה ומחליפה את `num` בשם הקובץ לערך `num` המתאים.  
כל זוג ייכתב בשורה נפרדת, עם האינדקס המתאים לשורה ולאחריו נקודה ורווח,  
ולאחר מכן שני המספרים הראשוניים מופרדים ביניהם על ידי מקף (-).  
ניתן להניח כי `num` מספר חיובי שלם גדול מ-0.

לדוגמא, הקריאה

`twin_pricks(4)`

תייצר קובץ בשם `q2a_output_4.txt` אשר יכיל את התוכן הבא :

```
1. 3-5
2. 5-7
3. 11-13
4. 17-19
```

הערות:

ניתן ומומלץ להשתמש בפונקציית עזר.  
תזכורת – המספר הראשוני הראשון הוא 2.

ב.

במשחק הילדים "7 בום!" המשתתפים צריכים לנקוב במספרים טבעיים בסדר עולה, אך בכל פעם שמגיעים למספר שמתחלק ב-7 או שמופיעה בו הספרה 7 יש לצעוק "boom!". כמו כן, אם המספר גם מתחלק ב-7 וגם מכיל את הספרה 7 (למשל המספר 7) יש לצעוק "boom-boom!". הגרסה המוכללת של המשחק נקראת "k בום!" וההגדרה שלה זהה לנ"ל כאשר k הינו מספר שלם כלשהו בין 1 ל-9 כולל.

ממשו את הפונקציה `K_boom(n,k)` אשר כותבת לתוך קובץ בשם `q2b_output_k_boom_n.txt` את כל המספרים בין 1 ל-n כולל בהתאם לחוקי המשחק "k בום!" כאשר n ו-k הם פרמטרים לפונקציה. בפורמט המתואר בדוגמא. ניתן להניח כי הפרמטרים n,k הם מספרים שלמים וגדולים מ-0.

לדוגמא : עבור הקריאה K\_boom(15,7) הפונקציה תיצור קובץ בשם q2b\_output\_7\_boom\_15.txt ותכתוב לתוכו בכל שורה בנפרד את המספרים מ-1 ועד 15 כולל לפי חוקי המשחק :

```
1
2
3
4
5
6
boom-boom!
8
9
10
11
12
13
boom!
15
```

### שאלה 3

ממשו את הפונקציה decipher\_rotate(in\_file) טקסט מוצפן מהקובץ שנמצא בנתיב in\_file , ומפענחת אותו על פי החוקיות המתוארת להלן :

עליכם לפענח את הטקסט שבקובץ הקלט על ידי החלפת כל אות אנגלית בשיטת המראה ולאחר מכן רוטציה של 5 מקומות אחורה : המראה של האות הראשונה היא האות האחרונה, ורוטציה של 5 אותיות לפניה תוביל לאות השישית מהסוף וכן הלאה (ראו דוגמאות בהמשך). אות גדולה (uppercase) תוחלף בגדולה ואות קטנה (lowercase) בקטנה. כל תו שאינו אות באנגלית (כולל רווחים וירידות שורה) יש להשאיר בדיוק כפי שהוא בקובץ הקלט. נתייחס בשאלה לא"ב כמחזורי, כלומר האות z היא האות שמגיעה לפני האות a.

דוגמאות : האות a תחילה תוחלף במראה שלה z ולאחר מכן באות החמישית לפניה כלומר ב-u.  
באותו אופן b תוחלף במראה שלה y ולאחר מכן באות החמישית לפניה שהיא t.  
האות z תוחלף באות a שהיא המראה שלה ולאחר מכן 5 תווים אחורה נקבל את האות v.  
דוגמה נוספת – l תוחלף ב-o שהיא המראה שלה ולאחר מכן ב-j שהיא 5 אותיות לפניה, ו-u תוחלף תחילה במראה שלה שהיא f ולאחר מכן ב-a שהיא 5 אותיות לפניה. באותו אופן בדיוק נחליף גם אותיות גדולות:

A → U  
B → T  
L → J  
U → A

לדוגמא, הטקסט המוצפן :

Fwbnggh mc pah!

Python is fun!

יפוענח ל-

והטקסט המוצפן :

Ynw rgh'b ckqjqbghe pmonb qusn gbnqd? Bnqw rgh'b nuzq bnq oabc!

יפוענח ל-

Why don't skeletons fight each other? They don't have the guts!

הפונקציה תכתוב את הטקסט המפוענח לקובץ באותה ספרייה בה נמצא קובץ הקלט. שם הקובץ המכיל את הטקסט המפוענח יוגדר לפי הכלל הבא :

הפונקציה **תוסיף** את המחרוזת "`_decipherotate`" לשם הקובץ המקורי לפני הנקודה האחרונה בקובץ (זו המציינת את סיומת הקובץ).

לדוגמא, בהנתן `in_file` בשם :

"C:\\path\\to\\file\\Q3\_input.txt"

שם הקובץ המכיל את הטקסט המפוענח יקרא (ההדגשה לצורך ההדגמה בלבד) :

"C:\\path\\to\\file\\Q3\_input\_**decipherotate**.txt"

דוגמא נוספת :

"C:\\path\\to\\file\\Q3.input.txt"

שם הקובץ המכיל את הטקסט המפוענח יקרא (ההדגשה לצורך ההדגמה בלבד) :

"C:\\path\\to\\file\\Q3.input\_**decipherotate**.txt"

**על הפונקציה להחזיר את שם הקובץ המכיל את הטקסט המפוענח.**

**הבהרות:**

- שימו לב שאתם מקפידים לסגור את הקובץ בכל תרחיש.
- ניתן להניח כי קיימת סיומת לקובץ הקלט. הסיומת אינה חייבת להיות .txt
- בהינתן שם קובץ על הנתיב אבסולוטי (מלא) כקלט, הפונקציה תחזיר את שם הקובץ המלא (כולל הנתיב מלא). בהינתן נתיב רלטיבי, יוחזר שם הקובץ החדש עם הנתיב הרלטיבי.

**שאלה 4**

ממשו את הפונקציה

`analyze_student_scores(students_ids,students_scores)`

המקבלת כקלט שני קבצי טקסט, קוראת את המידע בקבצים ומחזירה מילון כמו שיפורט למטה.

`students_ids` הוא קובץ טקסט, אשר מכיל בכל שורה `id,name` של סטודנט אחר בפורמט הזה :

```
101,John
102,Jane
103,Emily
104,Yossi
105,Ruth
106,Bob
```

`students_scores` הוא קובץ טקסט, אשר מכיל בכל שורה `id,score` של סטודנט אחר בפורמט הזה, כאשר עבור כל `id` ייחודי יכולות להיות מספר שורות של ציונים :

```
104,88
102,50
103,100
101,79
102,94
105,33
103,85
104,82
```

ניתן להניח כי נתיבי הקבצים חוקיים וקיימים וכי אין `id` של סטודנט שלא מופיע בקובץ `students_ids`, והמזהה (`id`) הוא ייחודי לכל סטודנט.

על הפונקציה להחזיר מילון כך שהמפתחות במילון יהיו `tuples` בפורמט (`id,name`) עבור כל סטודנט, והאיברים במילון יהיו רשימת הציונים שלו. במידה ואין עבור ציונים, נרצה שהערך יהיה רשימה ריקה. ה`ids` והשמות יהיו מטיפוס `str` והציונים יהיו מטיפוס `int`.

לדוגמא, עבור הקלטים שתוארו למעלה מילון הפלט הרצוי יהיה :

```
{('101','John'): [79], ('102','Jane'): [50, 94], ('103','Emily'): [100, 85], ('104','Yossi'): [88,82], ('105','Ruth'): [33], ('106','Bob'): []}
```

שימו לב כי סדר הזוגות במילון הוא לא קריטית אבל הסדר הפנימי ב`tuples` ובתוך הרשימות הוא כן קריטי. ב`tuples` יוגדר האיבר הראשון להיות המזהה והאיבר השני להיות השם. סדר ההכנסה לרשימות יהיה לפי הסדר הכרונולוגי של השורות בטקסט.