

תרגיל בית 5

הנחיות כלליות:

- קראו **היטב** את השאלות והקפידו שהתכניות שלכם פועלות בהתאם לנדרש.
- את התרגיל יש לפתור לבד!
- הקפידו על כללי ההגשה המפורסמים באתר. בפרט, יש להגיש את כל השאלות יחד בקובץ `ex5_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 הקלט/קבצי הטקסט 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"`

התרגיל מניח את קיומם של מספר אינסופי של אנשים ששםם ריק (Rick). לכל ריק ישנו **מזהה יחודי** (מזהה ריק) המורכב מהתווים הבאים :

1. מספרים
 2. אלפאבית אנגלי (אותיות) ,
 3. תווים מיוחדים : התווים הם מקף (-) וקו תחתי (_).
- מזהה **חייב** להסתיים בסיומת מספרית (הכוללת ספרות בלבד), **אך אינו חייב** להכיל תווים מיוחדים או אותיות מהאלפאבית האנגלי.

דוגמא למזהים לא-תקינים :

מזהה	מדוע לא תקני
C-137c	C מופיע בסוף המזהה
<u>bLndr</u>	אין מספר בסיומת
"C-137"	לא מסתיים בספרה, בנוסף ישנם תווים לא חוקיים (גרשיים)
137 "C"	תווים לא חוקיים (גרשיים)
	מחרוזת ריקה, לא נגמר במספר
137ריק	תווים לא חוקיים (אותיות בעברית)

דוגמא למזהים תקינים :

מזהה	תחילית טקסטואלית	סיומת מספרית
C-137	C-	137
bLndr151	<u>bLndr</u>	151
b1v3	b1v	3
29		29

שאלה 1- סכימת מספרים המופיעים בקובץ (rixum)

ממשו את הפונקציה rixum :

rixum(file_name)

הפונקציה מקבלת פרמטר יחיד בשם file_name, המכיל מחרוזת המציינת שם של קובץ קלט. הניחו שבתוך הקובץ מופיעה שורה בודדת המכילה סדרת מזהי ריק **חוקיים** המופרדים על ידי רווח בודד. על הפונקציה לקרוא את הקובץ ולהחזיר את סכום הסיומות המספריות של המזהים המופיעים בו.

עבור קובץ קלט המכיל את השורה הבאה :

rk2 c55 Q3 wWw67 rick10

הפונקציה תחזיר את הערך 137 (2+55+3+67+10).

בשאלה זו ניתן להניח כי קובץ הקלט קיים ותקין ואין צורך לטפל בשגיאות.

שאלה 2- ספירת ריקים (rickounter)

ברצוננו לספור את כמות מזהי ריק בקובץ טקסט נתון. אתם מתבקשים לממש את הפונקציה rickounter :

rickcounter(f_document, f_rick_identifiers)

הפונקציה מקבלת שני פרמטרים :

1. f_document : קובץ טקסט
2. f_rick_identifiers : קובץ המייצג סט של מזהי ריק. כל שורה בקובץ היא מזהה.

בהפעלת הפונקציה על נתיבי הקלט f_document, f_rick_identifiers, הפונקציה תחזיר את המילון הבא : המפתחות יהיו המזהים שנמצאים בקובץ f_rick_identifiers והערכים יהיו מספר המופעים שהופיע כל מזהה בטקסט שבקובץ f_document.

הבהרות:

- בשאלה זו ניתן להניח כי קבצי הקלט קיים ותקין ואין צורך לטפל בשגיאות.
- שימו לב שאתם מקפידים לסגור את הקובץ בכל תרחיש (כפי שנלמד בתרגול).
- מזהים רגישים לאותיות קטנות וגדולות. כלומר המזהה c7 שונה מהמזהה C7.
- מזהה חייב להתחיל ולהסתיים באותה השורה בקובץ f_document.
- מזהה יכול להופיע כחלק ממחרוזת ללא רווחים
- מזהה יכולים לחפוף אחד את השני
- ניתן להניח כי אין מזהה אשר מופיע פעמים בקובץ f_rick_identifiers.
- ניתן להניח כי הקובץ מכיל רק אלפאבית אנגלי, מספרים, מקף (-), קו תחתי (_), רווח וירידת שורה.
- במידה ומזהה כלשהו מהקובץ f_rick_identifiers אינו מופיע ב-f_documents, עליו להופיע כמיפויי במילון עם הערך 0.

דוגמא : בהנתן הקבצים הבאים (מודגשים רק המזהים שמופיעים ב-f_documents) :

f_rick_identifiers:

```
Hello_word9
CoolRick11
C-137
c-132
Z0Zo0
TestMeRick123
12rick12
```

f_document:

```
Hello world
CoolRick11 ImRick C-137 yeah Rick C-137 not rick c-132
Z0Zo0Out0 guy is lame CheckOut0 TestMeRick123
33333333ddd12rick12rick12rick
a
```

פלט הפונקציה יהיה המילון הבא:

```
{'Hello_word9': 0, 'CoolRick11': 1, 'C-137': 2, 'c-132': 1,
'Z0Zo0': 1, 'TestMeRick123': 1, '12rick12': 2}
```

מצורפים לתרגיל הקבצים q2_f_document_1.txt q2_f_rick_identifiers_1.txt כבדיקה לפונקציה. מומלץ ליצור קבצי בדיקה נוספים למקרי קצה בעצמכם.

שאלה 3 -תאומים ראשוניים (twin pricks)

ריק משער כי קיימים אינסוף זוגות של ראשוניים אשר ההפרש ביניהם הוא 2 בלבד. 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

ממשו את הפונקציה `twin_pricks(num, out_file_name)` אשר כותבת לתוך

הקובץ `name_file_out` את `num` זוגות התאומים הראשוניים הראשונים. כל זוג יכתב

בשורה נפרדת, כאשר שני המספרים יופרדו ביניהם על ידי פסיק (,).

לדוגמא, הקריאה

```
twin_pricks(4, 'q3_output_1.txt')
```

תייצר את הקובץ `q3_output_1.txt` אשר מכיל את התוכן הבא:

3,5

5,7

11,13

17,19

הנחיות:

- אם `num` קטן או שווה ל 0, עליכם לזרוק שגיאה מסוג `ValueError` עם ההודעה

הבאה:

Illegal value num={num}

כאשר `{num}` יוחלף בערכו של `num`.

העזרו בפקודה:

```
raise ValueError(f'Illegal value num={num}')
```

בשביל לזרוק את השגיאה עם המלל הנכון (הפקודה מופיעה גם בגוף שלד התרגיל)

- אם נזרקה שגיאת IO בעת הכתיבה לקובץ `name_file_out` עליכם לתפוס (except)

אותה ואז לזרוק (raise) שגיאת `ValueError` עם ההודעה הבאה:

Could not write to {out_file_name}

כאשר {out_file_name} יוחלף בשם הקובץ.

שאלה 4 – פענוח צופן (Rickode)

ממשו את הפונקציה rickode(in_file) הקוראת טקסט מוצפן מהקובץ in_file , ומפענחת אותו על פי החוקיות המתוארת להלן :

עליכם לפענח את הטקסט שבקובץ הקלט על ידי החלפת כל אות אנגלית באות **השניה** העוקבת **באופן מעגלי** (ראו דוגמאות בהמשך). אות גדולה (uppercase) תוחלף בגדולה ואות קטנה (lowercase) בקטנה. כל תו שאינו אות באנגלית (רווחים וירידות שורה) יש להשאיר בדיוק כפי שהוא בקובץ הקלט.

לדוגמא, a תוחלף בc ; f תוחלף בh ; B תוחלף בD. שימו לב כי האות y תוחלף בz ו Z תוחלף בb, ובאופן דומה תוחלפנה גם האותיות הגדולות (uppercase) : Y תוחלף בA ו A תוחלף בB. לדוגמא, הטקסט המוצפן :

“Rfcpc'X y jcqqml fcpc, ylb G'k lmr emgle rm zc rfc mlc rm dgespc gr msr”
יפוענח ל-

“There'Z a lesson here, and I'm not going to be the one to figure it out”.

הפונקציה תכתוב את הטקסט המפוענח לקובץ **ותחזיר את שם הקובץ המלא** (כולל הנתבי). שם הקובץ יוגדר לפי הכלל הבא :

הפונקציה תוסיף את המחרוזת “_decipherick” לשם הקובץ המקורי לפני הנקודה האחרונה בקובץ (זו המציינת את סיומת הקובץ).

לדוגמא, בהנתן in_file בשם :

C:\path\to\file\Q4_input.txt

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

C:\path\to\file\Q4_input_**decipherick**.txt

דוגמא נוספת :

C:\path\to\file\Q4.input.txt

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

C:\path\to\file\Q4.input_**decipherick**.txt

הבהרות:

- בשאלה זו יש לטפל בכל חריג IO שעולה :
 - במקרה כזה התכנית תזרוק (raise) שגיאת ValueError עם התוכן הבא :

Could not decipher {in_file} due to an IO Error

כאשר {in_file} יוחלף בשם קובץ שניתן כקלט

- שימו לב שאתם מקפידים לסגור את הקובץ בכל תרחיש (כפי שנלמד בתרגול).
- ניתן להניח כי קיימת סיומת לקובץ הקלט