

תרגיל בית 2

לולאות ורשימות

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 `ex2_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 (עז לרמת הרווח).**
- אופן ביצוע התרגיל : בתרגיל זה עליכם להשלים את הקוד בקובץ השלד המצורף.
- **אין לשנות את שמות המשתנים שכבר מופיעים בקובץ השלד של התרגיל.**
- יש לעבוד עם המשתנים שמופיעים בשלד התרגיל. על הקוד של כל שאלה לעבוד ולספק את התוצאה הדרושה עבור קלט שיוזן במשתנים שמופיעים בשלד. יחד עם זאת, אתם רשאים להוסיף משתנים נוספים ושונים מהמשתנים שמופיעים בקובץ השלד כראות עיניכם.
- **אין למחוק את ההערות שמופיעות בשלד.**
- **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).

הערה כללית – בשלב הזה של הקורס ניתן להניח שהקלט שלכם תקין מבחינת טיפוסים – כלומר, אם נתון לכם שמשתנה כלשהו הוא רשימה של מספרים שלמים, אין צורך לוודא את זה. מעבר לזה (למשל, האם הרשימה יכולה להיות ריקה או לא), לא ניתן להניח שום הנחה לגבי הקלט חוץ ממה שמצויין בשאלה.

שאלה 1:

בהינתן רשימה בשם lst הכוללת מספרים שלמים וחיוביים (מטיפוס int) ומספר שלם וחיובי כלשהו a (גם הוא מטיפוס int),

הדפסו למסך את האינדקס של המספר השלישי ברשימה המתחלק ב-a ללא שארית. במידה ולא קיים מספר כזה, יש להדפיס -1.

לדוגמה, עבור a=2 והרשימה:

lst = [1, 2, 3, 4, 5, 6, 7]

יודפס:

5

ועבור אותה רשימה ו-a=4, יודפס:

-1

הערות: ניתן להניח ש a שונה מ 0, אך lst יכולה להיות גם רשימה ריקה. ניתן להניח שהאינדקס גדול או שווה ל-0.

שאלה 2:

בהינתן רשימה של מחרוזות (איברים מסוג str) בשם lst2, מצאו את אורך המחרוזת הממוצע ברשימה והדפסו למסך את מספר המחרוזות שאורכן גדול ממש מאורך המחרוזת הממוצע. בצעו זאת באמצעות לולאת while וגם באמצעות לולאת for (שימו לב שצריך לפתור את השאלה פעמיים בעצם. פעם רק עם while ופעם רק עם for) ניתן להניח ש lst2 היא רשימה המכילה לפחות מחרוזת אחת.

לדוגמא:

עבור הרשימה:

lst2 = ['rick', 'and', 'morty']

אורך המחרוזת הממוצע הוא 4.

הפלט של התכנית יהיה:

The number of strings longer than the average is: 1

The number of strings longer than the average is: 1

כאשר כל שורה מייצגת שימוש ב-while או for.

שימו לב כי הממוצע לא בהכרח יצא שלם

שאלה 3:

בהינתן רשימה של מספרים אשר נסמנה ב-lst3, כתבו קטע קוד שמדפיס את סכום ההפרשים (בערך מוחלט) של כל זוג מספרים צמודים ברשימה.

מקרי קצה:

- עבור רשימה ריקה הדפסו 0.
- עבור רשימה המכילה מספר בודד, הדפסו 0.

לדוגמה, עבור הרשימה:

lst3 = [0, 1, 4, 3]

הדפסו:

5

$$|0-1| + |1-4| + |4-3| = 1+3+1=5 \text{ שכן}$$

שאלה 4:

נתונה רשימה של מספרים שלמים חיוביים בשם lst4 בעלת לפחות 2 איברים. כתבו קוד אשר מדפיס רשימה חדשה המבוססת על הרשימה lst4 ומקיימת את התכונה הבאה: **המכפלה בערך מוחלט** בין כל זוג איברים שכנים ברשימה החדשה, יהיה גדול ממש מכל המכפלות (בערך מוחלט) שקודמו לו. הרשימה החדשה תכיל את אותם המספרים כמו הרשימה המקורית, ובאותו הסדר. במידה ומספר כלשהו המופיע ב-lst4 לא עומד בתנאי ההפרש בין כל זוג מכפלות צמודות, מספר זה לא יופיע ברשימה החדשה.

לדוגמא, עבור הרשימה:

lst4 = [1, -4, 8, 3, -9, 10]

יודפס:

[1, -4, 8, 10]

הסבר: המכפלה (בערך מוחלט) בין 4- ל-8 הוא 32, והמכפלה בין 8 ל-3 הוא גם 24 ולכן 3 לא מופיע. לאחר מכן מושוות המכפלה (בערך מוחלט) בין 3 ל-9- אשר גם הוא אינו גדול מ-32 ולכן גם 9- לא מופיע. המכפלה (בערך מוחלט) בין 9- ל-10 היא כבר 90 אשר עומד בתנאי ולכן 10 מופיע.

עבור :

```
lst4 = [1, 2, 4, 8]
```

יודפס :

```
[1, 2, 4, 8]
```

הסבר : כל הרשימה עומדת בתנאי ולכן כולה תודפס.

דוגמא נוספת,

```
lst4 = [1, 3, 0, 2]
```

יודפס :

```
[1, 3]
```

הסבר : המכפלה בין 1 ל 3 נותנת 3. שאר מכפלות נותנות 0 ולכן 0, ו 2 לא מופיעים.

שימו לב שניתן להשתמש ב abs במידה וצריך. הפונקציה abs פולטת את הערך המוחלט של הקלט. הסבר לגבי הפונקציה ניתן לקבל כאן : <https://docs.python.org/3/library/functions.html>, או באמצעות הקלדת help(abs) ב-SHELL.

שאלה 5:

ערך ASCII הוא ערך מספרי שלם המשוך לכל תו כלשהו. נתונים מחרוזת my_string ומספר שלם k. כתבו קוד אשר מדפיס את תת המחרוזת באורך k המופיעה ב my_string אשר מורכבת מאותיות זהות **בעלות ערך ה ASCII הגבוה ביותר**. תת מחרוזת באורך k הינה סדרה רציפה של k תווים בתוך המחרוזת. ניתן להניח ש-k הוא מספר חיובי הגדול מ-0, וכן כי כל תווים ב my_string הם באלפבית האנגלי (הכולל אותיות קטנות וגדולות).

לקריאה נוספת על ערכי ASCII ראו : <https://he.wikipedia.org/wiki/ASCII>

דוגמאות :

עבור k = 1, my_string = "abaadddefggg", יודפס :

For length 1, found the substring g!

עבור k = 3, my_string = "abaadddefffgg", יודפס :

For length 3, found the substring fff!

עבור k = 2, my_string = "abcccccd", יודפס :

For length 2, found the substring cc!

במידה ולא נמצאה תת מחרוזת באורך k , תודפס ההודעה על כך שלא נמצאה מחרוזת באורך המתאים:

לדוגמא, עבור $k = 9$, `my_string = "abaadddefggg"`, יודפס:

Didn't find a substring of length 9

עבור מחרוזת ריקה וא כלשהו תמיד לא נמצא תת מחרוזת

רמז 1: היזכרו איך מייצרים בפשטות מחרוזת שהיא תו מסוים המופיע k פעמים ברצף.

רמז 2: הזכרו כיצד ניתן מי מבין 2 תווים כלשהם הוא בעל ערך ה ASCII גבוה יותר.

שימו לב כי יש הבדל בין אותיות גדולות לקטנות. למשל, אם נתונה המחרוזת "aaA" אז עבור $k=3$ לא תמצא מחרוזת מתאימה כי התת מחרוזת הכי ארוכה של אותיות זהות היא "aa" והיא באורך 2. במילים אחרות, שימו לב כי ישנו הבדל בין ערכי ה ASCII של אותיות גדולות וקטנות.