חשיבה מחשובית ותכנות בשפת פייתון
תרגיל בית 5

הנחיות כלליות:

- קראו **היטב** את השאלות והקפידו שהתכניות שלכם פועלות בהתאם לנדרש.
- את התרגיל יש לפתור לבד!
- הקפידו על הוראות ההגשה המפורסמות במודל. בפרט, יש להגיש את כל השאלות יחד בקובץ ex#_012345678.py המצורף לתרגיל, לאחר החלפת ה# במספר התרגיל והחלפת הספרות 012345678 במספר ת.ז שלכם, כל 9 הספרות כולל ספרת הביקורת. למשל, אם מספר תעודת הזהות שלי הינו 112233445 ואני מגיש את תרגיל מספר 1, שם הקובץ שאגיש יהיה ex1_112233445.py.
- 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 (כן כן, עד לרמת הרווח).**
- אופן ביצוע התרגיל: בתרגיל זה עליכם להשלים את הקוד בקובץ המצורף.
- **יש לעבוד עם המשתנים שמופיעים בשלד התרגיל ואין לשנות את שמם.** על קטע הקוד של כל שאלה לעבוד ולספק את התוצאה הדרושה עבור קלט שיוזן במשתנים שמופיעים בשלד (המשתנים שלידם סימני שאלה ומחכים לקלט כפי שמופיע בדוגמאות בתרגול). יחד עם זאת, אתם רשאים להוסיף משתנים נוספים כראותם עינכם.
- מומלץ להתעדכן בפורום לגבי שאלות של סטודנטים אחרים וכמובן, במידה ועדיין משהו לא ברור, לשאול בעצמכם.

<u>שימו לבים! (תקף החל מתרגיל בית מספר 5 ואילך)</u>

- ✓ שימו לב 1, החתימה של כל אחת מהפונקציות שעליכם לממש מופיעה כבר בשלד התרגיל ואין לשנותה. עליכם לממש את <u>גוף הפונקציה בלבד</u>. יש למחוק את הפקודה pass (היא נמצאת שם רק כדי שתוכלו להריץ חלקי קוד בלי לשים בהערה את המימושים החסרים) ולהחליף אותה במימוש המתאים.
- ✓ שימו לב 2, בשאלה בה נתבקשתם לממש פונקציה, אין לעשות שום דבר מעבר לכך. ניתן ואף רצוי לקרוא לפונקציה שכתבתם עם מגוון קלטים על מנת לוודא את תקינותה אך אין להשאיר את הקריאות הללו בהגשת התרגיל.
- ✓ שימו לב 3, אף פונקציה לא מדפיסה דבר אלא רק מחזירה ערך כלשהו.

<u>שאלה 1</u> –

ממשו את הפונקציה is_char_in_str(s1, c1) המקבלת מחרוזת s1 ותו נוסף c1 (גם הוא מטיפוס str כמובן). הפונקציה תחזיר True (כמשתנה מטיפוס bool) אם c1 מופיע בתוך s. הערך המוחזר חייב להיות מטיפוס bool (ולא מחרוזת או כל דבר אחר). להלן דוגמאות הרצה:

```
>>> x = is_char_in_str('banana', 'a')
>>> print(x)
True
>>> type(x)
<class 'bool'>
>>>
>>> x = is_char_in_str('banana', 'd')
>>> print(x)
False
>>> type(x)
<class 'bool'>
```

שאלה 2 –

ממשו את הפונקציה is_in_range(lst2,a,b) המקבלת רשימת מספרים lst2, ושני מספרים ממשיים a ו-b. הפונקציה תחזיר True (כמשתנה מטיפוס bool) אם כל האיברים ברשימה גדולים מ-a וגם קטנים מ-b. אם מספר אחד או יותר אינו עומד בקריטריון הנ"ל הפונקציה תחזיר False. הערך המוחזר חייב להיות מטיפוס bool (ולא מחרוזת או כל דבר אחר). ניתן להניח כי a<b. הבהרה: אם הרשימה ריקה יש להחזיר True (שכן היא עונה על הקריטריון הנ"ל באופן ריק). להלן דוגמאות הרצה:

```
>>> x = is_in_range ([1, 2, 3], 0, 5)
>>> print(x)
True
>>> type(x)
<class 'bool'>
>>>
>>> x = is_in_range([4.7, -1, -2], -2, 0)
>>> print(x)
False
>>> type(x)
<class 'bool'>
```

שאלה 3 – ממשו את הפונקציה upper_list_strings(lst3) שמקבלת רשימה של מחרוזות, ומחזירה רשימה המכילה את אותן המחרוזות ובה הוחלפו כל האותיות הגדולות באותיות גדולות. להלן דוגמאות הרצה:

```
>>> my_list = upper_list_strings(['this', 'is', 'A', 'TeST', 'caSe', '!'])
>>> print(my_list)
['THIS', 'IS', 'A', 'TEST', 'CASE', '!']
```

שאלה 4 –

ממשו את הפונקציה log10_list(lst4) שמקבלת רשימה של מספרים חיוביים, ומחזירה רשימה המכילה log בבסיס 10 של כל אחד מהמספרים ברשימת הקלט. אין "לייפות" את התצוגה. הדפיסו את תוצאות פעולת הlog כפי שהן. להלן דוגמאות הרצה:

```
>>> my_list = log10_list([10, 100, 1000, 10000])
>>> print(my_list)
[1.0, 2.0, 2.9999999999999996, 4.0]
>>> my_list =  log10_list([1, 2, 3, 4, 5])
>>> print(my_list)
[0.0, 0.30102999566398114, 0.47712125471966244, 0.6020599913279623,
0.6989700043360187]
```

שאלה 5 –

ממשו את הפונקציה is_palindrom(s5) שמקבלת מחרוזת ומחזירה True (כמשתנה מטיפוס bool) אם s5 הינה פלינדרום case-sensitive, אחרת מחזירה False. הערך המוחזר חייב להיות מטיפוס bool (ולא מחרוזת או כל דבר אחר). **אין להפוך את המחרוזת** (בכל דרך שהיא). להלן דוגמאות הרצה:

```
>>> x = is_palindrom('banana')
>>> print(x)
False
>>> type(x)
<class 'bool'>
>>>
>>> x = is_palindrom('Anana')
>>> print(x)
False
>>> type(x)
<class 'bool'>

>>> x = is_palindrom('anana')
```

```
>>> print(x)
True
>>> type(x)
<class 'bool'>
```