

שיעורי בית 1

הנחיות:

1. עליכם להגיש קובץ פייתון (קובץ עם סיומת py) ששמו תעודת הזהות שלכם. לדוגמה, סטודנט שתעודת הזהות שלו היא 012345678 יגיש קובץ בשם 012345678.py.
2. בכל שאלה יהיה עליכם לממש פונקציה. את כל הפונקציות יש לממש בקובץ המוגש. חשוב לוודא ששמות הפונקציות יהיו זהות לנדרש בשאלה.
3. כדי להקל, מצורף קובץ עם סיומת txt שבו מופיעות דוגמאות לקריאה לפונקציות שעליכם לממש, ואת הפלט שמוחזר מהן. אנחנו ממליצים לוודא שהפונקציה שלכם מחזירה את הפלט הרצוי על הדוגמאות הללו.
4. הקוד שלכם ייבדק בנוסף על קלטים אחרים מעבר לדוגמאות שניתנו לכם, כדי להבין אם הקוד עובד או לא (במיוחד במקרי קצה).
5. מומלץ לפתור את השאלות בסדר עולה משתי סיבות:
א. חלק מהשאלות מתבססות על שאלות קודמות.
ב. השאלות מסודרות לפי סדר קושי עולה.
6. נפתח עבורכם במודל פורום שאלות לגבי הנחיות תרגיל הבית. אם אתם מפרסמים בו, בבקשה תכתבו בכותרת את מספר התרגיל ומספר השאלה, כדי שסטודנטים אחרים יוכלו ללמוד מהשאלות שלכם.
7. **באף שאלה אין להשתמש בפונקציות מובנות של פייתון שלא ראינו בתרגול.** אם יש ספק – מוזמנים לשאול בפורום.
8. **מבנה התרגיל:** התרגיל מכיל 10 שאלות, שכל אחת שווה 11 נקודות. כלומר, ניתן להגיע לציון 100 גם עם מספר טעויות. לא ניתן לעבור את הציון 100.

שאלה 1

ממשו פונקציה בשם **min_list2** שמקבלת שתי רשימות, ומחזירה את הרשימה הקצרה מהשתיים.
הערות:

1. אין להשתמש בפונקציה המובנית `min`.
2. אפשר להניח שאורך הרשימות שונה.

שאלה 2

ממשו פונקציה בשם **count_in_list** שמקבלת רשימה, ומשתנה נוסף `x`. הפונקציה תחזיר כמה פעמים מופיע המשתנה `x` ברשימה.

שאלה 3

ממשו פונקציה בשם **count_even_in_list** אשר מקבלת רשימת מספרים, ומחזירה את כמות המספרים הזוגיים ברשימה.

שאלה 4

ממשו פונקציה בשם **min_list_even** שמקבלת שתי רשימות של מספרים. הפונקציה תחזיר את הרשימה שיש בה פחות מספרים זוגיים. ניתן להניח שבכמות האיברים הזוגיים בכל רשימה שונה.
רמז: השתמשו בפונקציות משאלות קודמות.

שאלה 5

ממשו פונקציה בשם **factorial** אשר מקבלת מספר שלם (`int`) אי שלילי, ומחשבת את עצרת המספר.
(עצרת מספר היא מכפלת כל המספרים השלמים מ-1 ועד אליו, כולל. לדוגמה: 5 עצרת שווה $1 \cdot 2 \cdot 3 \cdot 4 \cdot 5$ כלומר 120)

שאלה 6

ממשו פונקציה בשם **odd_factorial** שמקבל מספר שלם (`int`) אי שלילי. הפונקציה תחזיר את מכפלת כל המספרים האי-זוגיים מ-1 ועד המספר, כולל. לדוגמה: עבור המספר 5 נחשב $1 \cdot 3 \cdot 5$, כלומר 15.

שאלה 7

ממשו מחדש את שאלה 2, הפעם בשם **count_in_list2**. במימוש זה, השתמשו בסוג אחר של לולאה (אם השתמשתם ב-`while`, כעת השתמשו ב-`for` ולהיפך).

שאלה 8

ממשו מחדש את שאלה 6, הפעם בשם **odd_factorial2**. במימוש זה, השתמשו בסוג אחר של לולאה (אם השתמשתם ב-`while`, כעת השתמשו ב-`for` ולהיפך).

שאלה 9 (שאלה קשה יחסית)

ממשו פונקציה בשם **inverse_sum** אשר מקבל מספר שלם (int) בשם x, ומחזירה את המספר השלם הגדול ביותר, אשר סכום כל המספרים הטבעיים עד אליו קטן או שווה לא.
לדוגמה: עבור המספר 16 נקבל ש $1 + 2 + 3 + 4 + 5 = 15$, שקטן מ16.
עם זאת, $1 + 2 + 3 + 4 + 5 + 6 = 21$, שגדול ממש מ16. לכן, הפונקציה תחזיר 5.

שאלה 10 (שאלה קשה יחסית)

כתבו פונקציה בשם **inverse_odd_factorial** אשר מקבלת מספר שלם (int) בשם x, ומחזירה את המספר השלם הגדול ביותר, אשר הפלט של odd_factorial של משאלה 6 קטן או שווה לא.
לדוגמה: עבור המספר 16 מתקיים כי $1 \cdot 3 \cdot 5 = 15$ שקטן מ16, אבל $1 \cdot 3 \cdot 5 \cdot 7$ כבר גדול מ16. לכן, הפונקציה תחזיר 6.