

## עבודת הגשה מס' 1

### הנחיות:

- יש להגיש את העבודה עד **06/04/2025**
- על כל יום איחור בהגשה ללא הצדקה ירדו 3% מהציון. לא ניתן להגיש כלל באיחור של מעל שבוע.
- תוכניות יש להגיש בקובץ עם סיומת **.PY**.
- חשוב מאוד לציין את שם המגיש בתחילת הקובץ.
- אין להשתמש בנושאים שטרם נלמדו.
- על הקובץ להכיל דוגמאות קלט/פלט.
- יש לתעד (docstrings) כל התוכנית.
- פתרון שלא יעבוד בהרצה עם בגרסאות **python 3.11-3.13** – יקבל 0. בדקו היטב שאין שגיאות **syntax**.
- ההגשה היא באתר **moodle**. את העבודה יש להגיש על ידי סטודנט עם שם הקובץ שיהיה מורכב מהמילה "**HW1**" ומספרי ת"ז מופרדים בקו תחתון ביניהם.  
לדוגמא: **HW1\_123456789.py**

(1)

כתבו פונקציה בשם **Q1** שמקבלת כפרמטר רשימה הכוללת: מספרים שלמים - חיובים ושלילים. לפונקציה להחזיר רשימה כאשר בהתחלה יהיו מספרים מרשימה המקורית עם סכום ספרות מינימלית ואחריהם יהיו מספרים עם סכום ספרות מקסימלית.

### דוגמאות:

**Q1**([7,-17,23,145,809,-14,93,-6,5,-377,78]) => [23, -14, 5, 809, -377]

**Q1**([23,-5]) => [23, -5, 23, -5]

(2)

כתבו פונקציה בשם **Q2** שמקבלת כפרמטרים מספר שלם ומחרוזת. המחרוזת בנויה בצורת **slicing** לרציפים הכוללת סוגריים ושלושה מספרים: אינדקסים וצעד. לפונקציה להחזיר מספר שהיא מקבלת כפרמטר אחרי פעולת "**slicing**".

### דוגמאות:

**Q2**(-1234567,'[1:-1]') => -23456

**Q2**(-1234567,'[3:]') => -4567

**Q2**(1234567,'[:5]') => 12345

**Q2**(-1234567,'[1:-1:2]') => -246

**Q2**(1234567,'[::3]') => 147

**Q2**(-1234567,'[::-1]') => -7654321

**Q2**(-1234567,'[-2:0:-2]') => -642

**Q2**(1234567,'[2:10:2]') => 357

(3) בהינתן רשימת ציוני מבחנים. הרשימה מכילה ציונים בטווח 0-100, וכן ציון 199 עבור סטודנטים שלא ניגשו למבחן. ציון המעבר (אחרי פקטור) הוא 55.

(א) מרצה החליט לבדוק כמות סטודנטים עוברים כאשר משתמשים בפקטור שורש  $(10\sqrt{g})$ . השלימו קוד פונקציה כך אשר תקבל רשימת ציונים היא תחזיר כמות סטודנטים העוברים אחרי פקטור. יש ליצור שתי פונקציות אחד עם שם Q3a לפי מבנה pipeline ופונקציה בשם Q3a1 כאשר מבטלים על אחד מהשלבים של pipeline.

def Q3a(grades):

return reduce(<4>, filter(<3>, map(<2>, filter(<1>, grades))), <5>)

דוגמה:

Q3a([46, 199, 77, 100, 199, 14]) ==> 3

(ב) מרצה החליט לבדוק עבור חישוב פונקציות שונות אם כל הסטודנטים שניגשים למבחן מקבלים ציון עובר ואף סטודנט לא מקבל ציון יותר מ-100. השלימו קוד פונקציה כך, אשר תקבל רשימת ציונים ופונקציה לחישוב פקטור, היא תחזיר True עם כל הסטודנטים מקבלים ציון עובר ואף סטודנט לא מקבל ציון יותר מ-100 אחרי פקטור, אחרת False.

def Q3b(grades, func):

return reduce(<3>, map(<2>, filter(<1>, grades)), <4>)

דוגמאות:

Q3b([46, 199, 77, 100, 199], lambda x: x+15) ==> False

Q3b([46, 199, 77, 199], lambda x: x+15) ==> True

Q3b([46, 199, 77, 100, 199, 20], lambda x: x+15) ==> False

(4) השלימו פונקציה ללא שם בפונקציה Q4 אשר מקבלת שני פרמטרים רצף שכולל פונקציות ללא שם של פרמטר אחד שמחזירות ערך בוליאני (True/False) ורצף מספרים שלמים. לפונקציה להחזיר רצף המורכב ממספרים אשר רק אחת מהפונקציות בוליאניות מרשימת הפונקציות מחזירה ערך אמתי (True).

```
def Q4(funcs,data):
    return (lambda f, d: <1>)(funcs, data)
```

**דוגמה:**

```
Q4((lambda x: x>0, lambda x: x%2==0, lambda x: 9<abs(x)<100),(20,-45,133,8,400,7,
-300,68))
(-45, 133, 7, -300)
```

(5) שאלה זאת מתבססת על טיפוסים מובנים, פונקציות map, filter, reduce, Generator expressions, פונקציות מובנות של Python ופונקציות ללא שם בלבד. בכל סעיף אתם מתבקשים להשלים פונקציה ללא שם אשר מקבלת רשימת זוגות (tuple) המורכבות משם של סטודנט וציון שהוא קיבל. עבור כל סעיף קיימת דוגמה להרצה.

(א) השלימו פונקציה ללא שם כך שהוא תחזיר True אם כל הסטודנטים מרשימה קיבלו ציון עובר.

```
>>> marks=[('David',80),('Tali',45),('Moshe',86),('Yanir',80)]
>>> Q5a(marks) #False
>>> Q5a(marks[:1]+marks[2:]) #True
>>> Q5a=lambda x: <1>
```

(ב) השלימו פונקציה ללא שם כך שהוא תחזיר רשימת סטודנטים שלא קיבלו ציון עובר.

```
>>> Q5b(marks)
['Tali']
>>> Q5b(marks+[('Ran',50)])
['Tali', 'Ran']
>>> Q5b=lambda x: <2>
```

(ג) השלימו פונקציה ללא שם כך שהוא תחזיר רשימת סטודנטים עם ציונים אחרי פקטור מקסימלי עבור אותה רשימה.

```
>>> Q5c(marks)
[('David', 94), ('Tali', 59), ('Moshe', 100), ('Yanir', 94)]
>>> Q5c(marks+[('Ran',95)])
[('David', 85), ('Tali', 50), ('Moshe', 91), ('Yanir', 85), ('Ran', 100)]
>>> Q5c=lambda x: < 3 >
```

(ד) תשלימו פונקציה ללא שם כך שהוא תחזיר True אם קיימים שני סטודנטים ברשימה עם אותו ציון.

```
>>> Q5d (marks) #True
>>> Q5d (marks[:-1]) #False
>>> Q5d =lambda x: < 4 >
```

**בהצלחה !!!**