



## עבודת הגשה מס' 3

### הנחיות:

- **תאריך הגשה:** יש להגיש את העבודה עד לתאריך **04/01/2026**.
- **איחור בהגשה:** על כל יום איחור בהגשה ללא הצדקה, יירדו **3%** מהציון. לא ניתן להגיש כלל לאחר יותר משבוע איחור.
- **דרישות קובץ:** יש להגיש את התוכניות בקובץ עם סיומת **.py**.
- **שם המגיש:** חשוב מאוד לציין את שם המגיש בתחילת הקובץ בהערה (comment).
- **אין להשתמש בנושאים שטרם נלמדו.**
- **הקפדה על דוגמאות קלט/פלט:** הקובץ חייב לכלול דוגמאות קלט/פלט.
- **תיעוד:** כל תוכנית צריכה להיות מתועדת (Docstrings).
- **שגיאות תחביר:** פתרון שלא יעבוד בהרצה ב-IDLE בגרסאות Python 3.1x יקבל ציון 0. בדקו היטב שאין שגיאות תחביר (Syntax Errors).
- **שם הקובץ בהגשה:** יש להגיש את העבודה דרך אתר **Moodle**. שם הקובץ צריך להיות מורכב מהמילה "HW3" ומספרי תעודות הזהות של המגיש, מופרדים בקו תחתון. לדוגמה: **HW3\_123456789.py**.

### תרגילים:

**1** כתבו פונקציה בשם **minThree** שמקבלת כפרמטרים שלושה מספרים ממשיים ומחזירה את המספר הקטן ביותר ביניהם. הפונקציה תשתמש בפונקציה אחרת בשם **minTwo**, שמחזירה את המספר הקטן מבין שני מספרים שהיא מקבלת כפרמטר. (אין להשתמש ב-**if** בפונקציה **minThree**).

**דוגמאות:**

```
>>> minThree( 1, 4, 2 )
1
>>> minThree( 4, 2, 2 )
2
>>> minThree( 4, 1, 7 )
1
```

**2** כתבו פונקציה בשם **arrangeDigits** שמקבלת מספר שלם וחיובי, ומחזירה מספר חדש שבו כל הספרות הזוגיות מופיעות בהתחלה, והספרות האי-זוגיות מופיעות בסוף, תוך שמירה על סדרן המקורי. הפונקציה תשתמש בפונקציות עזר הבאות:

- **lastDigit:** מחזירה את הספרה האחרונה של המספר.
- **withoutTheLastDigit:** מחזירה את המספר ללא הספרה האחרונה.
- **isEven:** מחזירה True אם המספר זוגי, אחרת False.
- **addDigit:** מוסיפה ספרה למספר.



### דוגמאות:

```
>>> arrangeDigits(12345)
24135
>>> arrangeDigits(46842)
46842
>>> arrangeDigits(1375)
1375
>>> arrangeDigits(13574628)
46281357
>>> arrangeDigits(12547638)
24681573
```

(3) כתבו פונקציה בשם **printPyramid** שמקבלת מספר שלם וחיובי **n**, ומדפיסה פירמידת מספרים וסימני # בגובה **n**. הפונקציה תשתמש בפונקציה בשם **printChars**, שמקבלת תו ומספר ומדפיסה את התו באותו מספר של פעמים.

### דוגמה:

```
>>> printPyramid(9)
#
#1#
#212#
#32123#
#4321234#
#543212345#
#65432123456#
#7654321234567#
#####
```

(4) כתבו פונקציה בשם **sumNums** שמקבלת מספר שלם וחיובי **n**, וקולטת **n** מספרים שלמים תלת-ספרתיים. הפונקציה תחזיר את סכום המספרים אשר הספרה האמצעית שלהם היא ספרה זוגית. הפונקציה תשתמש בפונקציות אנונימיות (**lambda functions**) למשימות הבאות:

- להחזיר את הספרה האמצעית של מספר תלת-ספרתי.
- לבדוק אם מספר הוא זוגי.
- להחזיר את סכום שני מספרים.

### דוגמה:

```
>>> sumNums( 10 )
Enter 10 Triple digits number:
number: 123
number: 410
number: 354
```

### מגמה להנדסת תוכנה



מבוא לשפת פייתון וקריפטוגרפיה  
מיכאל פינקלשטיין

```
number: 222
number: 978
number: 777
number: 555
number: 202
number: 111
number: 432
547
```

(5) כתבו פונקציה בשם **doGame** שמפעילה משחק זיכרון. המשחק יכלול שלבים בהם השחקן צריך לזכור מספרים שהולכים וגדלים באורך (בשלב הראשון, מספר תלת-ספרתי, ובכל שלב נוסף המספר יגדל באחת). לאחר ניקוי המסך, השחקן יתבקש לבצע אחת מארבע משימות:

- לחשב ולכתוב את סכום הספרות של המספר.
- לכתוב את המספר בסדר הפוך.
- לחשב ולכתוב את סכום הספרה הגדולה ביותר והקטנה ביותר.
- לכתוב את המספר אחרי הזזה מעגלית שמאלה לספרה אחת.

אם השחקן עונה נכון, יתווספו לו נקודות לפי כמות הספרות במספר, והוא יעבור לשלב הבא. אם הוא טועה, יופחתו הנקודות שצבר עד כה.

בסיום המשחק, יוצג הניקוד הסופי של השחקן והודעה על סיום המשחק. התוכנית תשתמש בפונקציה **randrange** מספריית **random** ליצירת מספר אקראי.

דוגמה להרצה:

```
>>> doGame( )
Level 1
The number: 459
Remember number and Press enter...

The sum of all digits is: 18

Level 2
The number: 5746
Remember number and Press enter...

The shift left number is: 7465

Level 3
The number: 12156
Remember number and Press enter...

The reverse number is: 65121
```

מבוא לשפת פייתון וקריפטוגרפיה  
מיכאל פינקלשטיין

Level 4

The number: 844502

Remember number and Press enter...

The reverse number is: 205448

Level 5

The number: 4068412

Remember number and Press enter...

The sum of max and min digits: 1

Game over

You win 18 points

**בהצלחה !!!**