

## עבודת בית מס' 1:

### הנחיות כלליות:

- קראו את כל ההוראות לגבי הגשת תרגילי הבית באתר הקורס.
- קראו את כל העבודה לפני שתתחילו לפתור אותה.
- תאריך פרסום: 04.04.22
- תאריך הגשת התרגיל: 14.04.22 בשעה 23:59.
- יתאפשר איחורי הגשות בסך כולל של 5 ימים עבור כל התרגילים בסמסטר, לתרגיל שלא יוגש במסגרת זמן זה, יינתן ציון 0.
- כתבו תיעוד (הערות) שמסביר את הקוד שלכם. **אסור לכתוב הערות בעברית!**
- שאלות בנוגע לעבודה יישאלו בפורום המתאים במודל או בשעות הקבלה.
- את הקוד ניתן לכתוב ב-pyCharm כפי שהוסבר בתרגולים, או ישירות באתר ההגשה, הגשת העבודות תהיה באתר ההגשה.
- השימוש בחבילות מוכנות של פייתון **אסור בהחלט** ויגרור ציון 0.
- על הפלטים להיות בדיוק כפי שמוגדרים בשאלות (ללא רווחים מיותרים).**

### מערכת הבדיקות וההגשה האינטרנטית:

במערכת האינטרנטית תוכלו לצפות בתרגילי הבית, לבצע בדיקות אוטומטית של הקוד למול קלטים מסוימים ולראות כיצד הקוד שלכם מתנהג למול קלטים אלו.

**ביטים** < חזרה לתרגילים יש לי בעיה

תיאור התרגיל

עורך קוד

בחירת שפה

Python

הורדת הקוד

עורך הקוד

קלט-פלט ובדיקת קלט

**עבודה 1 - תרגיל 2**

מאתם מזוודה ובה מיליון דולר במזומן, המזוודה מוגנת בסיסמא בעלת 3 ספרות. המזוודה תיפתח אך ורק לאחר הזנה של קוד המכיל 3 ספרות שסכומן זהה ל-3 הספרות המקוריות. ממשו תכנית בשפת פייתון המקבלת 2 רשימות, כל אחת מכילה 3 מספרים שלמים בין 0-9, התכנית תבדוק האם סכומן זהה:

- אם הסכום זהה בדיוק, התוכנית תחזיר "JACKPOT"
- אם בין הסכומים הפרש של 1 (+/-) התוכנית תחזיר "ALMOST"
- אם הסכומים שונים בהפרש של יותר מ-1, התוכנית תחזיר "JACK WHO"

#	קלט	פלט רצוי	פלט מצוי
1	[4,1,0],[2,3,0]	JACKPOT	JACKPOT
2	[2,6,2],[2,6,1]	JACK WHO	JACKPOT
3	[6,1,6],[2,3,1]	ALMOST	JACKPOT

רמז #01

רמז #02

רמזים

**בדיקת הקוד:**

בדיקת הקוד תתבצע בלחיצה על כפתור "בדיקה", המערכת תריץ את הקוד שרשמתם אל מול הקלטים המוגדרים ותציג בטבלת הקלט-פלט את הפלט המצוי אל מול הפלט הרצוי, פלט נכון ייצבע בסגול, יש לשים לב כי פלט שלא ייצבע בסגול אינו נכון, גם שגיאה בתו בודד תגרום לפלט שגוי.

**הגשת התרגיל:**

להגשת הקוד, יש ללחוץ על כפתור "הגשה סופית", עם הגשה מוצלחת תופיע הודעה המסמלת על כך. כמות ההגשות אינה מוגבלת, כך שההגשה האחרונה תהא ההגשה שתיבדק, אנו ממליצים להוריד את קובץ הקוד לצורך גיבוי (ע"י לחיצה על כפתור ההורדה).

The screenshot displays a programming environment with the following elements:

- Code Editor:** Contains the following Python code:
 

```
2 def question2(PIN, GUESS):
3     return "JACKPOT"
```
- Test Cases Table:**

#	קלט	פלט רצוי	פלט מצוי
1	[4,1,0],[2,3,0]	JACKPOT	JACKPOT
2	[2,6,2],[2,6,1]	JACK WHO	JACKPOT
3	[6,1,6],[2,3,1]	ALMOST	JACKPOT
- Submission Area:** Includes input fields for "רמז #01", "רמז #02", and a "סיומן הסבר" (hint icon) button.
- Annotations (in Hebrew):**
  - A purple box highlights the output "JACKPOT" for all three test cases, with an arrow pointing to the text "פלט התכנית ושגיאות יופיעו כאן" (The program's output and errors will appear here).
  - Two buttons at the bottom are labeled "הגשה סופית" (Final Submission) and "בדיקה" (Check).
  - An arrow points from the "בדיקה" button to the text "בדיקת קוד" (Code Check).
  - An arrow points from the "הגשה סופית" button to the text "הגשת הקוד" (Submitting the code).

**שאלה 1:**

המרחק הקצר ביותר בין שתי נקודות במרחב דו ממדי  $(x_1, y_1)$  ו- $(x_2, y_2)$  מחושב באופן הבא:

$$d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

ממשו תכנית בשפת פייתון המקבלת 4 מספרים שלמים בסדר הבא  $x_1, y_1, x_2, y_2$  המייצגים 2 נקודות במרחב ומחזירה את המרחק הקצר ביותר בין שתי הנקודות עם דיוק של 2 נקודות עשרוניות.

לדוגמא:

קלט	הנקודות שיתקבלו	פלט
4,6,7,9	(7,9) ו- (4,6)	4.24
0,4,11,1	(0,4) ו- (11,1)	11.40
5,3,2,2	(5,3) ו- (2,2)	3.16

תזכורת:  $x^2 = x \cdot x$

התוכנית תמומש בתוך פונקציה באופן הבא:

```
def question1(x1,y1,x2,y2):
```

```
    # enter your solution here
```

```
    return "Insert solution here" # delete this line before you solve.
```

על הפונקציה להחזיר מספר עשרוני (על ידי שימוש ב- return).

בדיקת קלט והגשה יש לבצע בפלטפורמה האינטרנטית.

**שאלה 2:**

מצאתם מזוודה ובה מיליון דולר במזומן, המזוודה מוגנת בסיסמא בעלת 3 ספרות. המזוודה תיפתח אך ורק לאחר הזנה של קוד המכיל 3 ספרות שסכומן זהה ל-3 הספרות המקוריות. ממשו תכנית בשפת פייתון המקבלת 2 רשימות, כל אחת מכילה 3 מספרים שלמים בין 0-9, התכנית תבדוק האם סכומן זהה:

- אם הסכום זהה בדיוק, התוכנית תחזיר "JACKPOT"
- אם בין הסכומים הפרש של 1 (+/-) התוכנית תחזיר "ALMOST"
- אם הסכומים שונים בהפרש של יותר מ-1, התוכנית תחזיר "JACK WHO"

לדוגמא:

קלט - PIN	קלט - GUESS	פלט
[4,1,0]	[2,3,0]	JACKPOT

## Introduction to Programming in Python

JACK WHO	[2,6,1]	[4,1,0]
ALMOST	[2,3,1]	[4,1,0]

ניתן להניח כי הקלט תקין - רשימה המכילה 3 מספרים שלמים וחיוביים בין 0-9.

התוכנית תמומש בתוך פונקציה באופן הבא:

```
def question2(PIN):
```

```
    # enter your solution here
```

```
    return "Insert solution here" # delete this line before you solve.
```

על הפונקציה להחזיר מחרוזת (על ידי שימוש ב- return).

בדיקת קלט יש לבצע בפלטפורמה האינטרנטית.

שאלה 3:

מונח הוא **פלינדרום**, אם לאחר המרה של כל האותיות לאותיות קטנות (lowercase) **והסרה של תווים מיוחדים ורווחים** ניתן לקרוא אותו בדיוק באותה הצורה מההתחלה לסוף ומהסוף להתחלה, לדוגמא המונח 'evil olive' הוא פלינדרום לאחר הסרת רווחים ותווים מיוחדים.

ממשו תוכנית בשפת פייתון שמקבלת מחרוזת אלפאנומרית (מספרים ואותיות) ובודקת ומחזירה:

- אם המחרוזת היא פלינדרום – “Palindrome”
- אחרת – “Not Palindrome”

לדוגמא:

קלט	פלט
"a@b!Ba"	Palindrome
"raCE a cAr"	Not Palindrome
"Yo, banana boy!"	Palindrome

\*לא ניתן להניח כי המחרוזת מכילה רק מספרים ואותיות, **מחרוזת ריקה היא פלינדרום**.

התוכנית תמומש בתוך פונקציה באופן הבא:

```
def question3(sentence):
```

```
    # enter your solution here
```

```
    return "Insert solution here" # delete this line before you solve.
```

על הפונקציה להחזיר מחרוזת (על ידי שימוש ב- return).

בדיקת קלט והגשה יש לבצע בפלטפורמה האינטרנטית.

**שאלה 4:**

ג'ורג' מאוד אוהב פיצה, הוא מסתובב ברחבי העולם בחיפוש אחר פיצריות עם תוספות חדשות שהוא עוד לא טעם. כשג'ורג' מגיע לפיצרייה, הוא רשאי לאכול כמות פיצות השווה **למחצית מכמות הפיצות** הקיימות, אבל **לא יותר מאחת מכל סוג** (כדי לנסות כמה שיותר סוגים).

**דוגמא 1:**

אם בפיצרייה מסוימת יש 6 פיצות [1,1,2,2,3,3] בשלושה סוגים שונים (1 = זיתים, 2 = עגבניות, 3 = בצל) ג'ורג' יכול לאכול  $6 / 2 = 3$  פיצות, שהן אחת מכל סוג ומחצית מכמות הפיצות המוצעות.

**דוגמא 2:**

בפיצרייה מסוימת יש 4 פיצות באופן הבא [1,1,2,3] (שתי פיצות עם זיתים, אחת עם עגבניות ואחת עם בצל), ג'ורג' יכול לאכול  $4 / 2 = 2$  פיצות בלבד – [1,2] או [2,3] או [1,3] שהן מחצית מכמות הפיצות המוצעות ולא יותר מאחת מכל סוג.

ממשו תכנית בשפת פייתון המקבלת כקלט רשימה של סוגי פיצות ומחזירה את **כמות הפיצות המקסימלית** שיכול ג'ורג' לאכול.

\*ניתן להניח כי n (כמות הפיצות) הוא תמיד מספר זוגי.

קלט	פלט
[1,1,2,2,3,3]	3
[1,1,2,3]	2
[1,2,3,4,5,6,7,8]	4

רמז: כמות הפיצות הקיימות ו- כמות הסוגים השונים.

התוכנית תמומש בתוך פונקציה באופן הבא:

```
def question4(pizzas):
```

```
    # enter your solution here
```

```
    return "Insert solution here" # delete this line before you solve.
```

על הפונקציה להחזיר מספר שלם (על ידי שימוש ב- return).

בדיקת קלט והגשה יש לבצע בפלטפורמה האינטרנטית.

**שאלה 5:**

מספר ראשוני ז'רמן **p** הוא מספר שעבורו מתקיימים התנאים הבאים:

- $p$  מספר ראשוני (מתחלק אך ורק במספר 1 ובעצמו)
- $2p+1$  מספר ראשוני

לדוגמא:

המספר 11 הוא מספר ראשוני ז'רמן מאחר ומתקיימים התנאים הבאים:

- 11 הוא מספר ראשוני
- $2p+1 = 23$  הוא גם כן מספר ראשוני

מספר ראשוני ז'רמן יכול להיות כזה **ממעלה ראשונה, שנייה וכן הלאה**, לדוגמא:

המספר 5 הוא ראשוני ז'רמן **ממעלה שלישית** מאחר ו-11 ( $2*5+1$ ) הוא ראשוני ז'רמן, גם המספר 23 ( $2*11+1$ ) הוא ראשוני ז'רמן והמספר 47 ראשוני אך אינו ראשוני ז'רמן.

לצורך התרגיל, נגדיר את מונח זה כ- "דרגת ז'רמן".

ממשו תכנית בשפת פייתון המקבלת מספר שלם number ומחזירה:

- -1 – אם המספר אינו ראשוני
- 0 – אם המספר ראשוני רגיל (ולא ראשוני ז'רמן)
- אחרת - את "דרגת הז'רמן" של המספר

ניתן להניח  $p < 100$ .

קלט	פלט
11	2
8	-1
89	5

התוכנית תמומש בתוך פונקציה באופן הבא:

```
def question5(number):
```

```
    # enter your solution here
```

```
    return "Insert solution here" # delete this line before you solve.
```

על הפונקציה להחזיר מספר שלם (על ידי שימוש ב- return).

בדיקת קלט והגשה יש לבצע בפלטפורמה האינטרנטית.

**בהצלחה! אוראל.**