

## עבודת בית מס' 3:

## הנחיות כלליות:

- קראו את כל ההוראות לגבי הגשת תרגילי הבית באתר הקורס.
- קראו את כל העבודה לפני שתתחילו לפתור אותה.
- תאריך פרסום: 16.05.22
- תאריך הגשת התרגיל: 29.05.22 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 העבודה יש לכתוב בעורך הקוד המקובל בקורס (PyCharm) הגשת העבודה תתבצע באתר המודל בתיקייה הייעודית.
- **קובץ ההגשה יהיה קובץ מסוג .py. כאשר כותרת הקובץ היא תעודת הזהות של הסטודנטית.**
- השימוש בחבילות מוכנות של פייתון **אסור בהחלט** ויגרור ציון 0.
- **על הפלטים להיות בדיקת כפי שמוגדרים בשאלות (ללא רווחים מיותרים).**

## בדיקות ידניות:

לתרגיל זה מצורף קובץ בדיקות ידני, בעזרתו תוכלו לבדוק ב-PyCharm את הפונקציות שתממשו בתרגיל. הפונקציות ירצו למול קלטים שהוגדרו מראש וידפיסו לקונסולה את הקלט שהתקבל, הפלט הצפוי והפלט שלכם באופן הבא:

```

main()
Run: Assignment 02 (1) x
C:\Users\oriel\PycharmProjects\pythonProject\venv\Scripts\python.exe "C:/Users/oriel/Desktop/Teaching/Python/2
Function: binary_to_decimal
Input: 1010 ← קלט לפונקציה
ExpectedOutput: 10 ← פלט צפוי לפונקציה
YourOutput: 10 ← פלט שהתקבל בפועל
isEqual: True ← האם הפלטים זהים (כולל סוג הערך המוחזר)
-----
Function: binary_to_decimal
Input: 111010
ExpectedOutput: 58
YourOutput: 10
isEqual: False

```

## הגשת התרגיל:

קובץ ההגשה יוגש במודל ויהיה קובץ פייתון (מסוג .py). כאשר כותרת הקובץ היא תעודת הזהות של הסטודנטית.

## הנחיות לעבודה:

עבודה זו היא עבודה בנושא רקורסיה, חלק מן הבעיות ניתן לפתור בקלות בלולאה, אבל נדרש פתרון רקורסיבי.

- ניתן לממש פונקציית מעטפת בכל מקום בו לא נכתב בפירוש אחרת.
- לפונקציית המעטפת יש לתת שם משמעותי.
- אין לשנות את החתימה של הפונקציה המקורית בקובץ השלד לתרגיל.
- אין לכתוב לולאות בעבודה כלל!
- אין להשתמש בפרמטרי ברירת מחדל בהגדרה של פונקציה.
- שימו לב כי לא ניתן להשתמש גם בפעולות sum, min, max, in על אובייקטים אטרקטיביים (כדוגמאת רשימה או טאפל) מאחר ואלו משמשות כלולאה אלא אם נאמר במפורש, כן ניתן להשתמש ב-min, max, עבור 2 משתנים בלבד.

**שאלה 1 :**

מספר ראשוני ז'רמן  $P$  הוא מספר ראשוני (המתחלק אך ורק ב-1 ובעצמו) שעבורו מתקיים  $2P+1$  גם כן ראשוני. לדוגמא :

המספר 5 הוא מספר ראשוני ז'רמן מכיון ש-5 הוא מספר ראשוני וגם 11 הוא מספר ראשוני.

עליכם לממש בעזרת רקורסיה, פונקציה המקבלת מספר  $P$  ומחזירה True אם המספר ראשוני ז'רמן ואחרת מחזירה False.

\*הערות : המספר 1 אינו מספר ראשוני.

**חתימת הפונקציה –  $\text{question1}(p)$**

## שאלה 2:

\*אין להשתמש במתודות להמרה אוט' בין בסיסים כדוגמאת-`int()`

א. בהינתן מחרוזות המייצגות מספר בייצוג בינארי, עליכם לממש פונקציה רקורסיבית אשר תבדוק האם המספר הינו מספר בינארי תקין – מספר בינארי תקין מוגדר כמספר בעל כמות ספרות המתחלקת ב-8 ללא שארית (8,16 וכו') ומכיל רק את הספרות '0' או '1'.  
לדוגמא:

עבור הקלט – '10010101' הפונקציה תחזיר `True`.  
עבור הקלטים – '1000' או '21101011' הפונקציה תחזיר `False`.  
**חתימה הפונקציה – `question2_a(num_str)`**

ב. בהינתן מחרוזות המייצגות מספר בייצוג בינארי, עליכם לממש פונקציה רקורסיבית אשר תבדוק אם המספר הינו מספר תקין (בשימוש בפונקציית `question2_a`, אם המספר תקין, תמיר את המספר לייצוג דצימאלי (בסיס 10), בהינתן ההנחות הבאות:

מספר, עבורו ה-MSB (הסיפרה השמאלית ביותר) היא '1' יומר ע"י שימוש בשיטת המשלים ל-2 באופן הבא:

ביצוע משלים ל-2, המרה לדצימאלי, הוספת מינוס מקדים. לדוגמא:

עבור הקלט – '10001100', נמיר את כל הספרות כך שיתקבל '01110011' ונוסיף 1 כך שיתקבל '01110100'. את המספר שיתקבל נמיר לייצוג דצימאלי (116) ונוסיף מינוס מקדים, כך שהפונקציה תחזיר את התשובה **-116** את התשובה יש להחזיר כ-`integer`.

עבור הקלט – '1001' הפונקציה תחזיר `None` (מספר בינארי לא תקין).

מספר, עבורו ה-MSB (הסיפרה השמאלית ביותר) היא '0' יומר באופן פשוט למספר (בכל דרך שתבחרו) לבסיס דצימאלי, לדוגמא:

עבור הקלט – '01000111' יוחזר המספר 71.  
**חתימת הפונקציה – `question2_b(num_str)`**

**ניתן להשתמש בכמה פונקציות עזר.**

**שאלה 3:**

בהינתן רשימה של מספרים שלמים (integers), חיוביים ויחודיים (כל מספר מופיע פעם אחת) בשם "lst" ומספר שלם n, עליכם לממש פונקציה רקורסיבית שמחפשת ומחזירה את כמות תתי הרשימות (subset), אשר בפעולת כפל של איבריהם ניתן להגיע בדיוק למספר n, ניתן לחזור על איברים ב-2 או יותר תתי רשימות, אין חשיבות לסדר.

לדוגמא:

בהינתן lst = [2,4,6,12,7,3,1] ו-n=12 יוחזר הפלט 6.  
מאחר וישנם 6 תתי רשימות [2,6], [4,3], [12,1], [12], [1,3,4], [1,2,6] שמכפלתם 12.

**חתימת הפונקציה – question3(lst, n)**

## שאלה 4:

בהינתן רשימה של מספרים שלמים (integers), שני שחקנים (player1 ו player2) משחקים משחק בעזרת הרשימה. השחקנים משחקים בתורות, כאשר player1 מתחיל, שני השחקנים מתחילים את המשחק בתוצאה 0. כל שחקן בתורו שולף איבר אחד מקצה מסוים של מהרשימה קרי, game\_list[0] או game\_list[len(game\_list)-1]. שליפת האיבר מקצרת את הרשימה באיבר אחד. את ערך האיבר הנשלף, מוסיף השחקן לתוצאה שלו, המשחק נגמר כאשר אין יותר איברים ברשימה. על הפונקציה לסכום את כמות האפשרויות לנצחון עבור כל אחד מהשחקנים, הפונקציה תחזיר True במידה ול-player1 יש יותר אפשרויות לנצח ואחרת תחזיר False כאשר סיום בתיקו נחשב ניצחון ל-player1. על הפונקציה לבדוק את כל האפשרויות הקיימות.

## חתימת הפונקציה – question4(game\_list)

דוגמאות:

game\_list = [1, 5, 2]

output: False

כל אפשרויות המשחק:

winner	Player2	Player1
-	5	1
Player2	-	2
winner	Player2	Player1
-	5	2
Player2	-	1
winner	Player2	Player1
-	2	1
Player1	-	5
winner	Player2	Player1
-	1	2
Player1	-	5

2 נצחונות ל-player1 ו-2 נצחונות ל-player2 לכן הפונקציה תחזיר False.

game\_list = [1, 14, 20, 35]

output: True

מהלך המשחק:

winner	Player2	Player1
	14	1
Player2	35	20

winner	Player2	Player1
	14	1
Player1	20	35
winner	Player2	Player1
	35	1
Player2	20	14
winner	Player2	Player1
	35	1
Player2	14	20
winner	Player2	Player1
	20	35
Player1	1	14
winner	Player2	Player1
	20	35
Player1	14	1
winner	Player2	Player1
	1	35
Player1	14	20
winner	Player2	Player1
	1	35
Player1	20	14

5 ניצחונות ל-Player1 ו-3 ניצחונות ל-Player2, הפונקציה תחזיר **True**

**שאלה 5:**

בהינתן מטריצה (רשימה דו מימדית) בשם "terrain" בגודל  $N \times M$  כאשר  $N$  מספר השורות ו- $M$  מספר העמודות במטריצה, המטריצה מכילה בכל מקום  $i, j$  את הערך 0 או 1 (integer) כאשר 0 מייצג "יבשה" ו-1 מייצג חלק מ"נהר".

נהר מורכב מסדרה (בכל גודל) של 1-ים סמוכים אופקית או אנכית (ללא אלכסונים), **מספר ה-1ים המרכיבים נהר מגדירים את אורכו**.

עליכם לממש פונקציה שתמצא את אורך הנהר הגדול ביותר במטריצה terrain בצורה רקורסיבית. שימו לב כי נהר עשוי להתפתל, ולהיות בצורת L לדוגמא. במילים אחרות, נהר אינו חייב להיות קו ישר אופקי או אנכי.

**בנוסף:** פתרון בעזרת ממואיזציה.

**חתימת הפונקציה – question5(terrain)**

לדוגמא - עבור המטריצה:

```
terrain = [[0, 1, 1, 1],
           [0, 1, 0, 0],
           [1, 1, 1, 0],
           [0, 0, 1, 1]]
```

0	1	1	1
0	1	0	0
1	1	1	0
0	0	1	1

יתקבל הפלט ← 9, אורכו של הנהר הארוך ביותר.

**דוגמאות נוספות:**

0	1	1	1
0	1	0	0
0	0	1	0
0	0	1	1

פלט -4

1	0	0	0
1	1	0	0
0	0	1	0
1	1	0	1

פלט -3

0	1	1	1
0	1	0	0
0	1	1	0
0	0	1	1

פלט -8

**בהצלחה! אוראל.**