

### משימות למעבדה מס' 3

#### (פונקציות רקורסיביות)

1. אנחנו יכולים להגדיר את הסכום מ-1 ועד  $x$  ( $1+2+3+\dots+x$ ) רקורסיבית עבור  $x \geq 1$ . השלימו את התוכנית הבאה בפייתון לחישוב הסכום  $1+2+3+4+5+6+7+8+9+10$ , בצורה

#### רקורסיבית:

```
#compute and print 1 + 2 + ... + 10
print(sum( 10 ))
def sum(x):
#you complete this function recursively
```

2. כתבו פונקציה רקורסיבית בשם **PrintBinary** המקבלת מספר שלם חיובי ומדפיסה אותו בפורמט בינארי.

לדוגמא:

**PrintBinary(8) => 1000**

3. כתבו פונקציה רקורסיבית המקבלת מספר טבעי (שלם וחיובי) ומחזירה כמות ספרות של המספר.

תפעילו את הפונקציה על המספרים: 15, 105, 15015.

4. כתבו פונקציה רקורסיבית בשם **RecDigitcheck** אשר מקבלת ספרה ומספר שלם ומחזירה כמות פעמים שהספרה מופיעה במספר.

לדוגמא:

**RecDigitcheck(2,1622723) => 3**

5. כתבו פונקציה רקורסיבית המקבלת מספר שלם וחיובי (המהווה מיקום בסדרת פיבונאצ'י) ומחזירה את המספר בסדרה פיבונאצ'י הממוקם במקום הזה. במידה והפרמטר לא חוקי (לא מספר שלם וחיובי) הפונקציה תחזיר הודעת שגיאה המפורטת של הבעיה.

א. כתבו תכנית הנעזרת בפונקציה הנ"ל.

ב. השתמשו בתוכנית שלכם לחישוב המספרים ה-10, 20, 30 וה-40 בסדרה.

ג. מדוע לוקח הרבה יותר זמן לקבל תוצאה עבור מספרי פיבונאצ'י גדולים יותר?

**6. משחק זיכרון בפייתון - כתובו פונקציה רקורסיבית לפי ההנחיות הבאות:**

- א.** ניקוד המשתמש מתחיל מאפס ועבור כל תשובה נכונה המשתמש יקבל ניקוד  $x$ , כאשר  $x$  הינו כמות הספרות במספר שהמשתמש זכר.
- ב.** בכל שלב המשתמש מקבל מספר ארוך יותר לזכור.
- ג.** כאשר המשתמש מרגיש מוכן הוא לוחץ על מקש ה-**ENTER** והמסך מתרוקן. לאחר-מכן, המשתמש יכול להזין את המספר שאותו הוא זוכר.
- ד.** אם התשובה שגויה, למשתמש יופיע על המסך הניקוד הסופי שלו והמשחק יסתיים.

**אקראיות (רנדומליות) בפייתון:**

```
import random
random.random() or random.randrange(10, 100)
```

**ניקוי מסך:**

```
print("\n" * 100)
```

**הפעלת פונקציה רקורסיבית:**

```
#new game - score=0, level 1: numbers in range 10-100
rec_memo(0, 1)
def rec_memo(score, level):
    #think different
```

- 7.** כתבו פונקציה רקורסיבית בשם **printFigure** המדפיסה פירמידת ספרות. הפירמידה נבנית באופן הבא: ראשית, לכל שורה חדשה הנוצרת מוסף מספר חדש שערכו שווה למספר השורה. בנוסף, עבור כל שורה אי-זוגית, יודפסו הערכים בסדר עולה ולעומת זאת עבור כל שורה שמספרה זוגי יודפסו הערכים בסדר יורד.

**לדוגמא:**

```
printFigure(9)
1
21
123
4321
12345
654321
1234567
87654321
123456789
```

**בהצלחה !**