**משימות למעבדה מס' 3**

**(פונקציות רקורסיביות)**

1. אנחנו יכולים להגדיר את הסכום מ-**1** ועד **𝑥** (1+2+3+⋯+𝑥) רקורסיבית עבור **𝑥 ≥ 1**. השלם את התוכנית הבאה ב**פייתון** 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

1. כתוב פונקציה רקורסיבית בשם **PrintBinary** המקבלת מספר שלם חיובי ומדפיסה אותו בפורמט בינארי.

**לדוגמא:**

**PrintBinary(8)** מדפיסה => **1000**

1. כתוב פונקציה רקורסיבית המקבלת מספר טבעי (שלם חיובי) ומחזירה כמות ספרות שלו.

יישם את הפונקציה על המספרים: **15**, **105**, **15015**.

1. כתוב פונקציה רקורסיבית בשם **RecDigitcheck** אשר מקבלת ספרה ומספר ומחזירה את מספר החזרות של אותה ספרה במספר.

**לדוגמא:**

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

1. כתוב פונקציה רקורסיבית המקבלת מספר שלם חיובי )המהווה מיקום בסדרת **פיבונאצ'י**) ומחזירה את המספר בסדרה **פיבונאצ'י** הממוקם במיקום זה. במידה והפרמטר לא חוקי (לא מספר שלם חיובי) הפונקציה תחזיר הודעת שגיאה המפורטת על הבעיה .
   1. כתוב תוכנית הנעזרת בפונקציה הנ"ל.
   2. השתמש בתוכנית שלך לחישוב המספרים ה- 10 , 20 , 30 וה-40 בסדרה.
   3. מדוע לוקח הרבה יותר לקבל תוצאה עבור מספרי **פיבונאצ'י** גדולים יותר ?
2. **משחק זיכרון בפייתון** - כתוב פונקציה רקורסיבית לפי ההנחיות הבאות:
   1. ניקוד המשתמש מתחיל מאפס ועבור כל תשובה נכונה המשתמש יקבל ניקוד **𝑥**, כאשר **𝑥** הינו כמות הספרות במספר שהמשתמש זכר.
   2. בכל שלב המשתמש מקבל מספר ארוך יותר לזכור.
   3. כאשר המשתמש מרגיש מוכן הוא לוחץ על מקש ה-**ENTER** והמסך מתרוקן. לאחר-מכן, המשתמש יכול להזין את המספר שאותו הוא זוכר.
   4. 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

1. כתוב פונקציה רקורסיבית בשם **printFigure** המדפיסה פירמידת מספרים, 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 !**