מבנה שפות תוכנה – 2

ביטויי למבדה ופונקציות על, closure

בכל התרגילים, אם לא נאמר אחרת, יש לכתוב בצורה הפונקציונלית הטובה ביותר.
בתרגיל זה נתרגל שימוש בפונקציות על מובנות של python וכתיבה של פונקציות כאלו.

1. כתבו פונקציה המבטאת את הפונקציה הלינארית הבאה Y = x/2+2 באמצעות למבדה
   1. עשו שימוש בביטוי הלמבדה על רשימה מ0-10000 וצרו רשימה חדשה ע"י פונקצית על.
   2. סכמו את איברי הרשימה בעזרת פונקציית על.
   3. השוו בין זמן הריצה של הסכימה באמצעות פונקציית העל לזמני הריצה של סכימה בשיטה אימפרטיבית.
   4. כעת, בצעו זאת באמצעות פונקציית על אחת בלבד שתסכום וגם תבצע את הקריאה ללמבדה.

2. צרו רשימה מ1-1000 וחלקו את הרשימה לשתי רשימות שונות כך שכל המספרים הזוגיים יהיו ברשימה אחת וכל האי זוגיים ברשימה השניה, השתמשו בפונקציית על.
   1. כתבו 2 פונקציות למבדה

   - פונקצית שתקבל מספר ותת-רשימה מתוך רשימת המספרים הזוגיים,

     כפלו את תוצאת מכפלת המספרים עד למקום הנוכחי ברשימה באיבר הבא ברשימה.
     - לדוגמא עבור הרשימה: 1,2,3,4,
       בצעד הראשון נכפיל 1*2 ובשני 2*3 ואחר כך 6*4 וכן הלאה.

   - עבור רשימת המספרים האי זוגיים, עשו שימוש בפונקציה הליניארית

     מלמעלה כאשר הX מייצג את הסכום עד לאיבר הנוכחי = x/2+2+next
     וnext הוא האיבר הבא ברשימה.
   2. הריצו את הפונקציה המתאימה עבור כל אחת מהרשימות בעזרת פונקציית על
   3. סכמו את התוצאה של כל רשימה בעזרת פונקצית על

3. א. כתבו פונקציה המקבלת תאריך לועזי בתור מחרוזת ו2 מספרים : הראשון הוא מספר התאריכים המבוקש והשני הוא מספר הימים לדילוג, הפונקציה תחזיר רשימה של התאריכים. יש להשתמש בפונקציות על.
   ב. (רשות) כתבו פונקציה דומה עבור תאריכים עבריים
4. א. כתבו פונקציה המקבלת את מעריך החזקה ומחזירה פונקציה המקבלת את בסיס החזקה ומחזירה את התוצאה.

לדוג' :

```
power_of_2 = power_function(2)

result_1 = power_of_2(3)  # 3^2) 9 (צריך להחזיר

result_2 = power_of_2(5)  # 5^2) 25 (צריך להחזיר
```

ב. כתבו פונקציה המקבלת מספר ומחזירה אובייקט map של פונקציות החזקה עד אותו מספר (לא כולל)

לדוג' עבור הקלט 5 יחזור אובייקט שמייצג את הרשימה : $[1,\ x^1,\ x^2,\ x^3,\ x^4]$

בנוסף, כתבו **סקריפט ראשי** המקבל מספר עבור החזקה הגבוהה ביותר מהמשתמש ומפעיל את הפונקציה

הסקריפט ידפיס את הטיפוס של האובייקט שהפונקציה החזירה

לאחר מכן, יקבל מספר עבור הבסיס X וידפיס tuple מתאים

לדוגמא עבור הבסיס 2 והחזקה 5 יודפס : $(1,\ 2,\ 4,\ 8,\ 16)$

נתון חלק מה**סקריפט הראשי** :

```
n = int(input("Enter number of powers:"))
 result = …..
print(type(result))
base = int(input("Enter base:"))
….
```

דוגמת הרצה :

```
Enter number of powers:
5
<class 'map'>
Enter base:
2
(1, 2, 4, 8, 16)
```

ג. כתבו פונקציה המקבלת מספר x וחזקה ומחזירה את קירוב טיילור עבור e^x ע"י שימוש בסעיפים הקודמים, במימוש הפונקציה אין להשתמש בלולאות או ברשימות

תזכורת : קירוב טיילור עבור e^x מוגדר ע"י הנוסחא הבאה $\mathrm{e}^x = \sum_{n=0}^{\infty} \frac{x^n}{n!} \quad \forall x$

כאשר סוכמים עד חזקה מסוימת.

5. מנהל משימות (closure)

כתוב **פונקציה** בשם task_manager שמאפשרת לנו לנהל משימות פשוטות באמצעות מילון. המפתח הוא שם המשימה והערך הוא מצב המשימה (incomplete,in progress, complete)

הפונקציה תחזיר מנהל משימות שמסוגל לבצע את הפעולות הבאות:

'add_task(task): הוסף משימה חדשה למילון שמייצג את המשימות. המשימה תתוסף עם מצב ברירת המחדל "incomplete", אלא אם נספק מצב אחר.

get_tasks(): החזר את המילון שמייצג את המשימות הנוכחיות.

complete_task(task): שנה את מצב המשימה להושלמה ("complete").

הפונקציה של מנהל המשימות תחזיר מילון שיאפשר גישה לפונקציות

דוגמה לשימוש במנהל המשימות :

```
# Create a new task manager
tasks_manager = task_manager()

# Add tasks
tasks_manager['add_task']("Write email")
tasks_manager['add_task']("Shopping", "in progress")
tasks_manager['add_task']("Homework")

# Get the list of tasks
current_tasks = tasks_manager['get_tasks']()
print(current_tasks)
# Should print: {'Write email': 'incomplete', 'Shopping': 'in progress', 'Homework':'incomplete'}

# Mark a task as complete
tasks_manager['complete_task']("Write email")

# Get the list of tasks after marking and deleting
current_tasks = tasks_manager['get_tasks']()
print(current_tasks)
# Should print: {'Write email': 'complete', 'Shopping': 'in progress', 'Homework': 'incomplete'}
```