

## עקרונות שפות תוכנה

תשע"ז, סמסטר א'

### עבודת הגשה מס' 2

#### הנחיות:

- יש להגיש את העבודה עד **18/12/2016**
- על כל יום איחור בהגשה ללא הצדקה ירדו 3% מהציון. לא ניתן להגיש כלל באיחור של מעל שבוע.
- מותר להכין את העבודה בזוגות או בודדים. חל איסור להעתיק או "לחלוק" חלקים מהפתרונות.
- תוכניות יש להגיש בקובץ עם סיומת **PY** ודיאגרמות בקובץ **PDF** מכווצים יחד ב- **RAR/ZIP**.
- חשוב מאוד לציין את שמות המגישים בתחילת הקבצים.
- חובה להשתמש בשמות הפונקציות המוגדרות.
- אין להשתמש בנושאים שטרם נלמדו (כגון מבני נתונים).
- על הקובץ להכיל גם פקודות הרצה של פונקציות.
- יש לתעד (docstrings) בגוף כל פונקציה.
- פתרון שלא יעבוד בהרצה ב-IDLE בגרסה python 3.5 – יקבל 0. בדקו היטב שאין שגיאות syntax.
- ההגשה היא באתר moodle. את העבודה בזוגות יש להגיש על ידי שני סטודנטים עם שם הקובץ **RAR/ZIP** זהה שיהיה מורכב מהמילה "**HW2**" ושני מספרי ת"ז מופרדים בקו תחתון ביניהם.  
לדוגמא: **HW2\_123456789\_123456789.zip**
- כל שאלה ופניה בנוגע לתרגיל יש להפנות אך ורק לאחראי על התרגיל – מיכאל פינקלשטיין  
בדוא"ל: [misha@tcb.ac.il](mailto:misha@tcb.ac.il) פניות בכל בדרך אחרת – לא יענו!

**(1)** צייר דיאגרמת סביבה עבור הפעלות פוקנציה הבאה. ציין מה יופיע בפלט:

```
x=5
def func1(x,y=2):
    def func2(z):
        global x
        def func3(z):
            nonlocal y
            z+=5
            print('1-{0},{1},{2}'.format(x,y,z))
            y+=z
            return z
        x+=func3(2)
        z=z-1
        print('2-{0},{1},{2}'.format(x,y,z))
        return z
    return func2(x)+2
func1(2,3)
func1(5)
```

**(2)** צייר דיאגרמת סביבה עבור הפעלות פוקנציה הבאה. ציין מה יופיע בפלט. (שימו לב שיש מספר הרצות שונות):

```
z = 0
def f(x, y):
    def g(x):
        nonlocal y
        global z
        z = z+1
        y = y-1
        x = x*y
        print("z={0}, y={1}, x={2}".format(z, y, x))
    return g
f = f(5, 10)
f(7)
for n in range(-1,1):
    f(n)
```

### חלק ב: פונקציות מסדר גבוה ופונקציות ללא שם (lambda)

**(3)** צייר דיאגרמת סביבה עבור הפעלות פונקציה הבאה. ציין מה יופיע בפלט:

```
def f1(x,f4):
    def f2(x):
        while (lambda x: x>10)(x):
            x=x//10
        return x
    def f3(x,y):
        return x==y
    x=f4(x)
    return f3((lambda y:y%10)(x),f2(x))
f1(11,lambda x:x**2)
f1(11,lambda x:x+7)
```

**(4) שיטת הטרפז** - מתאר שיטת אינטגרציה נומרית אשר משתמש בטרפזים בין כל שתי נקודות של הפונקציה בתור קירוב לפונקציה האמתית. בשיטה זאת אינטגרל מסוים של פונקציה  $f$  בין שני נקודות  $a$  ו- $b$  מקורב ע"י הנוסחה:

$$\int_a^b f(x) dx \approx \frac{h}{2} \sum_{k=1}^N (f(x_{k+1}) + f(x_k))$$
$$= \frac{b-a}{2N} (f(x_1) + 2f(x_2) + 2f(x_3) + \dots + 2f(x_N) + f(x_{N+1})).$$

עבור  $n+1$  נקודות במרווחים שווים בין  $a$  ל- $b$ , כל  $n$  זוגי, הנקודה ראשונה היא  $a$  ונקודה אחרונה היא  $b$ .

הגדירו פונקציה **Trapez\_rule** שמקבלת כארגומנטים הבאים:  $n$ ,  $a$ ,  $b$ ,  $f$  ומחזירה ערך של אינטגרל המחושב לפי שיטת הטרפז. ניתן להיעזר בפונקציה **summation** שמימשנו בכיתה.

**דוגמא:**

עבור אינטגרל  $x^9$  עם נקודה  $a=0$  ונקודה  $b=10$  ומספר  $n=100000$ :

`Trapez_rule(lambda x:x**9,0.0,10.0,100000)`

התוצאה היא 1000000000.75

**5** לעיתים נרצה להפעיל פונקציה על סדרה של איברים. לשם כך נשתמש בפונקציה מסדר גבוה המתפעלת פונקציות אחרות ורצף של נתונים (sequence).

כתוב את הפונקציות הבאות:

**א** פונקציה `myFilter(L,func)` - אשר מקבלת רצף (מסוג list) של מספרים ופונקציה כלשהי ומחזירה רשימה מתוקנת (יש להשתמש ב-list) שכוללת רק מספרים שהפונקציה מחזירה עבורם אמת.

**ב** פונקציה `myFilterMulti(L,funcL)` - אשר מקבלת רצף של מספרים ורצף של פונקציות ומפעילה סינון על סמך אותן הפונקציות על המספרים ברצף. פונקציה תחזיר רשימה (list) מתוקנת. הסינון יתבצע על סמך עמידה בעל התנאים יחד (וגם).

**ג** פונקציית עזר בשם `myPrime(x)` - תקבל מספר ותחזיר True/False האם הוא ראשוני.

**ד** פונקציית עזר בשם `isFib(x)` - תחזיר True/False אם x הוא חלק מסדרת פיבונצ'.

**דוגמא:**

ההפעלה הבאה: `myFilter([2,4,5,6],myPrime)`

תחזיר את הרשימה: `[2,5]`

הפעלה הבאה: `myFilterMulti([2,4,5,6,7,13],[myPrime,isFib])`

תחזיר את הרשימה: `[2,5,13]`

**ה** תפעיל פונקציה `myFilterMulti` כך שרצף פונקציות תכלול פונקציה ללא שם שבודקת שמספר הוא מספר דו-ספרתי.

**דוגמא:**

ההפעלה הבאה:

`myFilterMulti ([2,4,5,13,41,55,89,107,144],[myPrime,isFib, פונקציה ללא שם])`

תחזיר את הרשימה: `[13,89]`

**(6)** עבור פונקציה נומרית  $f$  ומספר שלם אי-שלילי  $n$ , ניתן להגדיר הפעלה ה- $n$ -ית של  $f$  כ- $n$  הפעלות של  $f$  על קלט מסוים. הגדירו פונקציה **repeated** אשר מקבלת פונקציה  $f$  ומספר  $n$  ומחזירה הפעלה ה- $n$ -ית של  $f$ .

**דוגמא:**

```
def square(x):  
    return x**2  
def incr(x):  
    return x+1
```

(להפעיל 4 פעמים פונקציה על ארגומנט)

```
>>> repeated (incr,4)(2)
```

**6**

(... או פעמיים ..)

```
>>> repeated (square,2)(5)
```

**625**

השתמשו בפונקציה `compose` שלמדנו:

```
def compose(f,g):  
    return lambda x: f(g(x))
```

*קראו בעיון את ההנחיות לתרגילים לפני שממחרים לשאול...*

**בהצלחה!**