



מטלה 2

מועד הגשה : 3/12/2020

- ✓ ניתן להכין את המטלה **בזוגות**
- ✓ רק **חבר אחד** בצמד **יגיש** בפועל את העבודה
- ✓ יש להגיש את הפתרון **תחת שם המכיל את מספרי ת"ז של כל המגישים**.
- ✓ יש להגיש קובץ פורמט PY.
- ✓ **חובה** להשתמש בשמות הפונקציות המוגדרות.
- ✓ שימו לב, הפלט של דוגמאות ההרצה הוא בהתאם לסביבת הפיתוח Python .IDLE
- ✓ חובה לכל פונקציה להוסיף doc strings .
- ✓ הגשה דרך **מודל בלבד!**
- ✓ כל שאלה בנוגע לתרגיל יש להפנות אך ורק לאחראית על התרגיל – גבי מאי חג'בי באימייל: mayha@ac.sce.ac.il . פניות בכל בדרך אחרת – לא יענו!
- בפנייה, יש לציין את : שם הקורס, שם הקמפוס ופרטים מזהים.
- ✓ **אישורי ההארכה** יינתנו ע"י מרצה בלבד !

* הערה חשובה: קיים הבדל עקרוני בין הדפסה לבין החזרה של ערך מפונקציה! ברירת המחדל בהיעדר הוראת הדפסה מפורשת היא **החזרה בלבד**.



המכללה האקדמית להנדסה סמי שמעון

חלק א: דיאגרמות סביבה

1. צייר דיאגרמת סביבה עבור הפעלות פונקציה הבאה. ציין מה יופיע בפלט:

```
x=15
def f1(x):
    def f2(x):
        return x*5
    return f2(x)
def f3(f4,y):
    return f4(y)*x
print(f3(f1,x+3))
```

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

```
k = 0
def f1(x, y):
    k = 1
    def f2(z):
        nonlocal y
        global k
        x = z
        k = k+1
        y = y+1
        print('k={0}, y={1}, x={2}'.format(k, y, x))
        x = x+1
    print('k =', k)
    y = x
    return f2
f1(5, 6)(7)
f1 = f1(8, 9)
for n in range(3):
    f1(n)
```



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

1. צייר דיאגרמת סביבה עבור הפעלות פונקציה הבאה. ציין מה יופיע בפלט:

```
def f1(x,f2):
    def f3(x):
        return x==0
    def f4(x):
        return x%10
    def f5(x):
        return x//10
    y=0
    while not f3(x):
        if f2(f4(x)): y = (lambda x:x+1)(y)
        x = f5(x)
    return y
print(f1(52,lambda x:x%2!=0 and x > 3))
print(f1(943,lambda x:x%3==0))
```

2. 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:

- a. פונקציה **my_filter(f)** אשר מקבלת פונקציה בוליאנית כארגומנט. הפונקציה תייצר מרשימה נתונה, רשימה חדשה שמכילה רק את האיברים עליהם f החזירה אמת.
- b. פונקציה **filtered_words** אשר מקבלת רשימה של מילים, words (מסוג list) ומספר שלם n . הפונקציה צריכה להחזיר רשימה של המילים שאורכם גדול מ- n . שימו לב - יש לממש פונקציה זו כפונקציית lambda. (ניתן להיעזר בסעיף a).
- c. פונקציה **my_map(lst, f)** אשר מקבלת פונקציה כארגומנט. הפונקציה תייצר מרשימה נתונה, רשימה חדשה המורכבת מערכי ההחזרה של הפונקציה f על האיברים ברשימה. כלומר, אם $lst = [x_1, x_2, x_3, x_4, \dots]$ אז נקבל $[f(x_1), f(x_2), f(x_3), f(x_4), \dots]$.
- d. פונקציה **my_filter_and_map(lst, f1, f2)** אשר מקבלת רשימה של איברים (lst), פונקציה בוליאנית $f1$ ופונקציה $f2$. הפונקציה תחזיר רשימה חדשה המורכבת מערכי ההחזרה של הפונקציה $f2$ אך רק מהאיברים עבורם $f1$ החזירה אמת. שימו לב - יש לממש פונקציה זו כפונקציית lambda. (ניתן להיעזר בסעיפים a ו-c).



המכללה האקדמית להנדסה סמי שמעון

דוגמאות הרצה :

```
>>> my_filter(lambda x: x%2 == 0)
<function my_filter.<locals>.lst_filter at 0x0000020B7F57A6A8>
>>> my_filter(lambda x: x%2 == 0) ([1, 3, 2, 5, 4, 8, 32, 41])
>>> [2, 4, 8, 32]
>>> my_filter(lambda x: x%2 != 0)([1, 3, 2, 5, 4, 8, 32, 41])
>>> [1, 3, 5, 41]
>>> filtered_words(['Hello', 'Hi', 'ABCDE', 'a', 'python'], 3)
['Hello', 'ABCDE', 'python']
>>> filtered_words(['Hello', 'Hi', 'ABCDE', 'a', 'python'], 1)
['Hello', 'Hi', 'ABCDE', 'python']
>>> my_map(lambda x: x*3)
<function my_map.<locals>.lst_map at 0x000002162BB9A6A8>
>>> my_map(lambda x: x*3) ([1, 3, 2, 5, 4, 8])
>>> [3, 9, 6, 15, 12, 24]
>>> my_filter_and_map(['Hello', 'Hi', 'ABCDE', 'a', 'python'], lambda w: len(w)%2 != 0,
lambda w: w[0])
['H', 'A', 'a']
>>> my_filter_and_map([2, 1, 8, 7, 45, 12, 15, 51], lambda x: x%2 == 0, lambda x: x**2)
[4, 64, 144]
```

3. כתבו פונקציה **polynomial(factors)** לחישוב ערך של פולינום בנקודה מסוימת אשר בהינתן **factors**, המתאר את הרשימה של ערכי הפולינום בסדר הבא $[a, b, c, \dots]$

$$ax^n + bx^{n-1} + cx^{n-2} + \dots$$

ערך הפולינום בנקודה מסוימת x .

דוגמת הרצה :

```
>>> parabola = polynomial([3, 5, 6])
>>> parabola
<function polynomial.<locals>.eval at 0x0000024506AFA1E0>
>>> value = parabola(3)
>>> value
48
>>> polynom = polynomial([1, 2, 1, 0])
>>> value = polynom(4)
>>> value
100
```



המכללה האקדמית להנדסה סמי שמעון

4. עליכם לכתוב משחק. המחשב מגריל מספר בעל 5 ספרות ומאפשר למשתמש לגלות מהו המספר, באמצעות שישה רמזים שהמחשב מעלה בצורה אקראית. בתום כל רמז ניתנת האפשרות להאזין את המספר. כל רמז ואפשרות הקלדת מספר מורידה 10 מ-100 נקודות התחלתיות. בכל שלב (לאחר רמז או ניצחון/הפסד) יש להציג למשתמש את מספר הנקודות שיש לו. במידה והמשתמש גילה את המספר הנכון מציגים לו את כמות הנקודות שהוא זכה. במידה שלא נשארו נקודות יש להדפיס הודעה ולסיים את המשחק. על המחשב להגריל את הרמזים באופן אקראי. הרמזים שינתנו למשתמש הם:

- א. תינתן למשתמש הזדמנות להקיש מספר והמחשב יחזיר לו האם המספר הסודי גדול יותר (או שווה) מהמספר שהוקש.
- ב. תינתן למשתמש הזדמנות להקיש מספר והמחשב יחזיר לו האם המספר הסודי קטן יותר (או שווה) מהמספר שהוקש.
- ג. סכום כל הספרות. לדוגמא יודפס: "sum: 15". חייבים להגדירה כפונקציה ללא שם (lambda).
- ד. חיסור כל הספרות (מהספרה הכי שמאלית לימנית). לדוגמא יודפס: "sub: -3". חייבים להגדירה כפונקציה ללא שם (lambda).
- ה. הצגת ספרות אי-זוגיות. לדוגמא יודפס: "odd digits: -XX--". הכוונה שכל סיפרה אי-זוגית מסומנת ב-"X" וכל סיפרה זוגית ב-"-".
- ו. הצגת ספרות שמתחלקות ב-3 ללא שארית. לדוגמא יודפס: "digits divided by 3: -X-X-X". הכוונה שכל ספרה המתחלקת ב-3 ללא שארית תסומן ב-"X" וכל סיפרה אחרת ב-"-".

עליכם לכתוב פונקציה בשם `code_cracker()` שמפעילה את המשחק. יש לחבר את הרמזים בהתאם להנחיות.

הנחיות ומבנה הפונקציה:

- אין להשתמש בפונקציות עזר מחוץ ל-`code_cracker()`
- יש ליצור פונקציה מסדר גבוה בשם `yes_or_no(f, n)` שמקבלת כארגומנטים פונקציה ומספר שלם n. הפונקציה תחזיר את המחרוזת 'Yes' במידה והפעלת הפונקציה f על המספר n החזירה אמת, אחרת תחזיר 'No'. (יש להשתמש בפונקציה עבור רמזים א' ו-ב').
- יש ליצור פונקציה `print_msg_to_func(msg, f)` שמקבלת כארגומנט מחרוזת (הודעה) ופונקציה, ומפעילה את f כדי לבצע הדפסה בהתאם לרמז. (יהיה שימוש בפונקציה הזאת רק עבור רמזים ג', ד')
- יש ליצור פונקציה `show_string_by_func(msg, f)` שמקבלת כארגומנט מחרוזת (הודעה) ופונקציה. הפונקציה תפעיל את f כמה פעמים שצריך ותחזיר את ההודעה msg יחד עם מחרוזת המורכבת מ-"X" ו-"-" בהתאם להפעלות של הפונקציה f. (יהיה שימוש בפונקציה הזאת רק עבור רמזים ה', ו')
- יש להיצמד לסגנון הקלט/פלט שמוצג ב-3 דוגמאות הרצה המצורפות.
- התרגיל יבדק לפי תקינות המשחק ע"פ הכללים שנקבעו וגם מקוריות המימושים. יש יותר מפתרון אחד "נכון".

SCE

```
>>>
welcome to the pin cracker game!
clue #1
sum : 13
points left: 90
guess or press ENTER for exit:
12343

clue #2
digits divided by 3 : XXX-X
points left: 80
guess or press ENTER for exit:
36310

clue# 3
sub : -7
points left: 70
guess or press ENTER for exit:

wrong, bye bye!
Points left: 70
>>>
```

```
>>>
Welcome to cracker game!
clue# 1
Enter number to check if the code is
smaller: 50000
Yes
points left: 90
guess or press ENTER for exit:40000

clue# 2
sum : 18
points left: 70
guess or press ENTER for exit:22653

yes, correct!
points left: 70
>>>
```

```
>>>
Welcome to cracker game!
clue# 1
sum : 22
points left: 90
guess or press ENTER for exit: 52555

clue# 2
digits divided by 3 : -X--X
points left: 80
guess or press ENTER for exit:10876

clue# 3
Enter number to check if the code is
smaller: 20000
No
points left: 70
guess or press ENTER for exit:70816
....
....
clue# 10
odd digits : -XXXX
points left: 0
guess or press ENTER for exit: 45698

wrong, bye bye!
points left: 0
>>>
```

בהצלחה !