

דף תרגילים מס' 2

- (1) כתבו פונקציה ללא שם שמקבלת שני פרמטרים x, y מספרים שלמים וחיובים, אם y מספר זוגי לפונקציה להחזיר x , אחרת שורש ריבועי של x .

לדוגמא:

4,6 – 4096

4,5 – 2

- (2) כתבו פונקציה ללא שם שמקבלת ערך מספרי ומחזירה 1 אם מספר חיובי, 2 אם מספר שלילי אחרת 3.

לדוגמא: 15 <- 1, -45 <- 2, 0 <- 3

- (3) בעזרת פונקציה `map` להחזיר כמות מספרים אי-זוגיים (שורה אחת).

לדוגמא:

4 – (23,12,35,1,0,486,721)

- (4) בעזרת פונקציה `map` להפוך סדר מספרים וסדר ספרות בכל מספר ברצף מספרים שלמים וחיובים (שורה אחת).

לדוגמא:

(123,45,678,9) – (9, 876, 54, 321)

- (5) בעזרת פונקציה `filter` ליצור רשימה מורכבת ממספרים תלת ספרתיים כשאר סכום ספרתם שווה ל-3 או ל-8 (שורה אחת).

אחת.

לדוגמא:

[102, 107, 111, 116, 120, 125, 134, 143, 152, 161, 170, 201, 206, 210, 215, 224, 233, 242, 251, 260, 300, 305, 314, 323, 332, 341, 350, 404, 413, 422, 431, 440, 503, 512, 521, 530, 602, 611, 620, 701, 710, 800]

- (6) בעזרת פונקציה `reduce` לחשב מכפלת מספרים דו-ספרתיים המתחלקים ל-8 (שורה אחת).

לדוגמא:

4114592413463347200

- (7) בעזרת פונקציה `reduce` להחזיר `True` אם כל המספרים של רצף מספרים זוגיים, אחרת `False`.

לדוגמא:

(21, 45, 6, 17, 58) – False

(2, 450, 634, 170, 58) – True

מבוא לשפת פייתון וקריפטוגרפיה 2
מיכאל פינקלשטיין

(8) נרצה לממש פעולה דומה ל-**reduce** אבל כללית יותר (לא מוגבלת לסכימת הערכים) – **accumulate**. זו פונקציה אשר מקבלת רצף, פונקציה שמקבלת שתי פרמטרים, וערך התחלתי.
לדוגמא:

```
>>> accumulate(0, add, (1, 2, 3, 4, 5))  
15  
>>> print(accumulate(1, lambda x,y: x*y, (1, 2, 3, 4, 5)))  
120  
>>> print(accumulate(1, lambda x,y: 4*x-y, (1, 2, 3, 4, 5)))  
571
```

(9) ממשו פונקציה **mymap** אשר מבצעת פעולה זהה ל-**map** של פייתון (עבודה עם tuple).
לדוגמא:

```
>>> tuple(map(lambda x: x*x, range(1, 5)))  
(1, 4, 9, 16)  
>>> mymap(lambda x: x*x, range(1, 5))  
(1, 4, 9, 16)
```

בהצלחה !!!