

1. כתוב בשורה אחת ביטוי המקבל רשימה של רשימות ומחזיר את מספר הרשימות בהן יש לפחות 2 איברים שהם מ TYPE שונה. יש לממש זאת באמצעות אחת או יותר מהפונקציות FILTER / REDUCE / MAP.

לדוגמא, עבור הקלט הבא:

```
[[1,5,3], ['a','v',3], ["sss", 'b'], [], [[3,4,5],['a']], [(4,5,6),[4,5,6]]]
```

יוחזר המספר 2.

2. השלם את השורה הרביעית בתכנית הבאה, כך שהפלט של הרצתה יהיה לוח הכפל, כמתואר מטה.

```
vlst = ['x'+str(num) for num in range(10)]
lamdic = {}
for i,l in enumerate(vlst):

    for v in vlst:
        for i in range(1,len(vlst)+1):
            print(lamdic[v](i),end=' ')
        print()
```

```
1 2 3 4 5 6 7 8 9 10
2 4 6 8 10 12 14 16 18 20
3 6 9 12 15 18 21 24 27 30
4 8 12 16 20 24 28 32 36 40
5 10 15 20 25 30 35 40 45 50
6 12 18 24 30 36 42 48 54 60
7 14 21 28 35 42 49 56 63 70
8 16 24 32 40 48 56 64 72 80
9 18 27 36 45 54 63 72 81 90
10 20 30 40 50 60 70 80 90 100
```

3. השלם את השורות החסרות בתכנית הבאה, על מנת שפלט התכנית יהיה כנתון מטה:

```
class a():

    def __init__(self,y):

    def _____:
        if z > self.y:
            return z-self.y
        else:
            return self.y-z
class b(_____):
```

```

def _____:
    if z > self.y:
        return z-self.y
    else:
        return self.y-z
print(a(5)(b(6)()))
print(a(6)(b(5)(6)))

```

הפלט:

3
5

4. כתוב פונקציה שמקבלת רשימה של מספרים לא ממוינים. התכנית תחזיר את הרשימה ממוינת, מבלי להשתמש בזכרון. כלומר אין להגדיר אף משתנה עזר או רשימת עזר. יש לממש את המיון באמצעות שימוש בפונקציות עזר או בביטויי lambda.

5. א. ממש דקורטור העוטף כל פונקציה שהיא. בכל קריאה לפונקציה עטופה הוא ידפיס סטטיסטיקה של מספר הקריאות הכולל לפונקציות עטופות ומספר הקריאות לפונקציה הנוכחית ומספר כולל של הפרמטרים (גם positional וגם keyword) שהועברו עד כה לפונקציה הנוכחית.

ב. הוסף להדפסה נתון לגבי כל הטיפוסים (TYPES) השונים שהוחזרו עד כה ע"י הפונקציות העטופות. למשל, אם היו 3 קריאות לפונקציות שהחזירו TUPLE ו-4 קריאות לפונקציות שהחזירו INT – תרשם הסטטיסטיקה הנ"ל בקריאה הבאה לפונקציה עטופה.

ג. האם לדעתך ניתן לממש את הדקורטור בסעיף א – בשפת JAVA? אם כן – ממש. אם לא – הסבר מדוע.

6. א. ממש באמצעות מחלקה דקורטור שבאמצעותו ניתן יהיה לחשב את הקלוריות בפיצות שונות. פיצה מורכבת מבצק, רוטב, גבינה ותוספות. סוגי הבצק: עבה, דק, ללא גלוטן עבה, ללא גלוטן רך. סוגי הרטבים: עגבניות, עגבניות עם שמנת, תרד עם שמנת, פסטו. סוגי הגבינה: מוצרלה, פרמזן, תחליף טבעוני. ניתן להזמין פיצה ללא גבינה או עם כמות גבינה רגילה או כמות גבינה כפולה. פיצה יכולה להיות ללא תוספות או עם עד 3 תוספות שונות על כל הפיצה או רק על מחציתה. התוספות האפשריות: זיתים, תירס, עגבניות, בצל. (תוכל לבחור את כמות הקלוריות לכל רכיב – כרצונך.)

ב. ממש את הדקורטור הנ"ל ב JAVA. (תוכל להיעזר בקישור הבא, המדגים מימוש של הדקורטור שראינו בכיתה באמצעות JAVA. <http://www.javabyexamples.com/decorator-pattern-in-java>).

7. השתמשו בתכונות Coroutine על מנת להכין 3 סבבים להפעלת 4 פונקציות.

פונקציה a() מיצרת מספר אקראי בין 1-5, ושולחת אותם לשתי פונקציות שממתינות למספר b2() ו b3().

פונקציה b2() מעלה את המספר האקראי בחזקת 2.

פונקציה b3() מעלה את אותו מספר אקראי בחזקת 3.

פונקציה c() ממתינה ל b2() ו ל b3(). מחברת בין התוצאות ומציגה את התוצרת, יחד עם המשתנה האקראי שנקבע בפונקציה a().