

שפות תכנות – תרגיל 3

1. א. כתוב פונקציה שמקבלת מספר שלם ומחזירה True אם סכום הגורמים שלו גדול או שווה למספר עצמו. (למשל, עבור המספר 28 אשר גורמיו הם 1,2,4,7,14 הפונקציה תחזיר True. עבור המספר 12 אשר מחלקיו הם 1,3,5 הם מחלקיו אשר 15 המספר עבור True תחזיר הפונקציה 1,2,3,4,6 False
ב. ממש בשורה אחת פונקציה (שורת ה return), המשתמשת בפונקציה בסעיף א. פונקציה תקבל list של מספרים שלמים וחייבים. הפונקציה תחזיר dict של כל הגורמים של כל המספרים המופיעים ב list ואשר סכום מחלקיהם גדול מהמספר עצמו. עבור כל גורם (key) – יצוין לכמה מספרים שונים הוא היווה גורם.
לדוגמה: עבור ה list: [12, 34, 28, 53, 64], המספרים המקיימים את התנאי שסכום גורמיהם גדול מהמספר הם 12 ו-28.
הגורמים של 12 הם: 1,2,3,4,6. הגורמים של 28 הם: 1,2,4,7,14, ולכן יוחזר המילון הבא:
{2:2, 3:1, 4:2, 6: 1, 7: 1, 14:1}
ג. ממש generator expression המקבל מספר, ובכל קריאה ל next – מחזיר את הגורם הבא של המספר. אם המספר ראשוני – הוא יחזיר false.
2. ממש בשורה אחת, תוך שימוש בביטוי lambda ובפונקציות map/reduce/filter, את הפונקציות הבאות:
א. פונקציה שמקבלת list של איברים מסוגים שונים ומחזירה dict שמפתחותיו הם: {'o': , 'f': , 'i': , 'c':}
עבור המפתח 'c' יופיע list הכולל את כל האיברים מסוג characters
עבור המפתח 'i' יופיע list הכולל את כל האיברים מסוג integer
עבור המפתח 'f' יופיע list הכולל את כל האיברים מסוג float
עבור המפתח 'o' יופיע list הכולל את כל האיברים מסוג אחר
ב. פונקציה שמקבלת רשימה של פונקציות שהוגדרו בתכנית, ומחזירה רשימה של כל הפונקציות שמועבר אליהן לכל היותר פרמטר אחד.
3. א. כתוב דקורטור שעוטף פונקציות המקבלות כקלט מספר מסוג int ומחזירות כפלט מספר מסוג int. הדקורטור ישמור tuple עם שני מספרים: הקלט הממוצע והפלט הממוצע. בכל קריאה לפונקציה שעטופה ע"י הדקורטור, יודפסו הערכים הממוצעים המעודכנים.
ב. כתוב דקורטור שעוטף פונקציות, ומייצר log של n הקריאות האחרונות. עבור כל קריאה ישמרו כל הקלטים שנשלחו אליה, והפלט שלה. הדקורטור יתמוך גם במשתני positional וגם ב keywords. לדוגמה:

```
>>> @logged
... def func(*args):
...     return 3 + len(args)
>>> func(4, 4, 4)
you called func(4, 4, 4) it returned 6
6
```
4. כתוב דקורטור שעוטף כל פונקציה שהיא, ובכל פעם שקוראים לפונקציה הוא מדפיס את שמות 3 הפונקציות האחרונות שעברו דרך הדקורטור. בפעמיים הראשונות – הוא לא ידפיס דבר.

5. השתמש ב- observable design patterns ו-observer וממש את המחלקה twitter כך שכל אובייקט יוכל לציין מסרים ולעקוב אחרי מסרים של אובייקטים אחרים. לאחר המימוש, ניתן יהיה להריץ את קטע הקוד הבא:

```
a = Twitter('Alice')
k = Twitter('King')
q = Twitter('Queen')
h = Twitter('Mad Hatter')
c = Twitter('Cheshire Cat')
a.follow(c).follow(h).follow(q)
k.follow(q)
q.follow(q).follow(h)
  h.follow(a).follow(q).follow(c)

print(f'==== {q.name} tweets ====')
q.tweet('Off with their heads!')
print(f'\n==== {a.name} tweets ====')
a.tweet('What a strange world we live in.')
print(f'\n==== {k.name} tweets ====')
k.tweet('Begin at the beginning, and go on till you
come to the end: then stop.')
print(f'\n==== {c.name} tweets ====')
c.tweet("We're all mad here.")
print(f'\n==== {h.name} tweets ====')
h.tweet('Why is a raven like a writing-desk?')
```