

שאלה 1

ממש פונקציה לפי הפרדיגמה הפונקציונלית שמקבלת מערך של מערכים המכילים מספרים שלמים ומחזירה את סכום המספרים.

לדוגמא, עבור המערך $[[12, 4], [1], [2, 3]]$, יוחזר המספר 22.

שאלה 2

סעיף א

כתבו generator function בשם hailstone שמחזירה את סדרת המספרים ב-hailstone sequence. הפונקציה מקבלת ארגומנט אחד שהוא המספר ההתחלתי בסדרה.

hailstone sequence היא סדרת מספרים המוגדרת כך. נסמן את המספר הנוכחי בסדרה ב-n.

אם n מתחלק בשתיים המספר הבא יהיה $n/2$ ואחרת המספר הבא יהיה $n*3 + 1$.

הסדרה מסתיימת כשמגיעים למספר 1 (קיימת השערה מתמטית שאומרת שעבור מספר התחלתי כלשהו הסדרה תמיד תגיע מתי שהוא למספר 1)

לדוגמא אם נריץ

```
(for i in hailstone(5):  
    print (i
```

אז יודפס

5

16

8

4

2

1

הערה חלוקה של שני מספרים מטיפוס int תניב תוצאה שהיא float (בגרסאות חדשות של python).

למשל התוצאה של $4/2$ היא 2.0 ניתן להמיר את התוצאה למספר שלם ע"י שימוש ב-int למשל

(2.0 int) נותן 2 או שאפשר להשתמש ב-floor division (יש אופרטור כזה המסומן //)

ב. ממש את סעיף א' תוך שימוש במחלקה.

ג. ממש את סעיף א'. באמצעות generator expresion

שאלה 3

ניתן לייצג גרף מכוון באמצעות מילון. למשל אם הגרף כולל את הקשתות

`a --> b, a --> c, b --> d, d --> a, e --> d`

אז ניתן לייצגו כך:

`= my_graph`

```
{ 'a': ['b', 'c'], 'b': ['d'], 'c': [], 'd': ['a'], 'e': ['d'] }
```

כתבו פונקציה `reachable(graph, node)` המחזירה את רשימת כל הצמתים שניתן להגיע אליהם מהצומת הנתון בגרף הנתון.

למשל, בהמשך לדוגמא הנ"ל אז `reachable(my_graph, 'a')` יחזיר `['a', 'b', 'c', 'd']`

הערה: ניתן לפתור את הבעיה ע"י ביצוע DFS או BFS. ניתן למצוא תיאורים של האלגוריתמים האלו באינטרנט (למי שרוצה תזכורת).

פסאודו קוד עבור שני האלגוריתמים ניתן למצוא למשל ב- Wikipedia

https://en.wikipedia.org/wiki/Breadth-first_search#Pseudocode

https://en.wikipedia.org/wiki/Depth-first_search#Pseudocode

כמובן שתצטרכו לעשות את ההתאמות הנדרשות ולא להשתמש באלגוריתמים בדיוק כפי שהם רשומים.

BFS עושה שימוש בתור (queue) בעוד ש-DFS (הגירסה האיטרטיבית) עושה שימוש במחסנית.

את שני מבני הנתונים האלו ניתן לייצג ע"י `list`. חפשו את הסעיפים

Using Lists as Stacks

Using Lists as Queues

ב- tutorial של `python` (<https://docs.python.org/3/tutorial/index.html>)

כדי לממש תור עדיף (מבחינת זמן ריצה) להשתמש ב- `collections.deque` (ראו בסעיף הנ"ל).

מאחר והגרף עלול להכיל מעגלים, שני האלגוריתמים (DFS, BFS) מחייבים לשמור בכל שלב את הצמתים שכבר

גילינו כדי שלא נטפל בהם שוב. ניתן לשמור את הצמתים שכבר גילינו בתוך `set`. זה עדיף על שמירתם

בתוך `list` כי חיפוש ב- `set` מהיר יותר (כי זה נעשה בעזרת hashing).

בסופו של דבר הפונקציה תחזיר `list` (ולא `set`).

שאלה 4

בהרצת התכנית הבאה:

```
var1 = [1, 2, 3]
var2 = (1, 2, 3)
print(id(var1), id(var2))
var1 += [4, 5]
var2 += (4, 5)
print(id(var1), id(var2))
```

התקבל הפלט הבא:

```
2098448316040 2098448339016
2098448316040 2098415186432
```

הסבר את הפלט.

שאלה 5:

א. משתנה מסוג int בפיתון הוא immutable. כתוב תכנית שמוכיחה את הטענה הזו.
ב. האם תוכל כתוב מחלקה בשם mutableInt שתחליף את int ובכל פעם שייצרו מופע של המחלקה הוא יהיה mutable. אם לא ניתן – הסבר מדוע. אם ניתן – ממש את המחלקה.

שאלה 6:

נתונות 3 פונקציות עזר ופונקציה נוספת sqrt הנתונה באופן חלקי. השלם את 3 השורות החסרות, כך שהפונקציה sqrt תחזיר את השורש הריבועי של מספר שלם, בדיוק של אלפיות (3 מקומות מימין לנקודה העشرונית).

```

def average(x, y):
    return (x + y)/2

def improve(update, close, guess=1):
    while not close(guess):
        guess = update(guess)
    return guess

def approx_eq(x, y, tolerance=1e-3):
    return abs(x - y) < tolerance

def sqrt(a):
    def sqrt_update(x):
        _____
    def sqrt_close(x):
        _____
    return _____

print(sqrt(256))

```

שאלה 7:

השתמש באחת או יותר מהפונקציות MAP / REDUCE / FILTER למימוש הסעיפים הבאים (אם יש צורך השתמש ב lambda expressions).

א. כתוב פונקציה המקבלת list של מספרים שלמים ומחזירה את הסכום של 2 בחזקת כל אחד מן המספרים החיוביים שקטנים מ - 12.

ב. ממשי dictionary של מצלמות. המפתח הוא string עם שם הדגם של המצלמה והערך הוא tuple המחזיק את שם היצרן, רזולוציה בפיקסלים ומחיר. לדוגמא:

```
cameras = { 'LEQ2B': ('Nikon', 3.68, 4995),  
            'CAE5D424105': ('Cannon', 3.40, 3899) }
```

כתוב פונקציה המחזירה את את הרזולוציה המכסימלית.

ג. בהתייחס לסעיף ב' – כתוב פונקציה המחזירה את הדגם של המצלמה בעלת הרזולוציה המכסימלית.