שאלה 1

א

.9

ב

התכנית מדפיסה את סכום הספרות של המספר

ג

הפקודה שנכשלה היא ההמרה למספר שלם כיוון שהקלט שהכנסנו הינו ביטוי

ד

470

ה

נתון שnum הראשון הוא מספר שלם חיובי ומכאן גם אי שלילי. אם כן הערך החדש שמוכנס לnum בכל איטרציה של הלולאה $\lfloor \frac{num}{10} \rfloor$ גם הוא אי שלילי ולכן הלולאה היא אינסופית.

שאלה 2

א

הפונקציה מקבלת מחרוזת עליה היא עובדת ומחזירה האם המחרוזת היא אלפאנומרית - מכילה רק מספרים ואותיות

```
str.isalnum("123abc")
-> True

str.isalnum("123-")
-> False
```

ב

הפונקציה מקבלת מחרוזת עליה היא עובדת, מחרוזת sep ומספר $maxsplit$ אופציונלי הפונקציה עוברת על המחרוזת, מחקלת אותה למחרוזות המופרדות ע"י sep ומחזירה רשימה של מחרוזות אלו. אם הפונקציה מקבלת את $maxsplit$ היא מחקלת את המחרוזת עליה היא עובדת מקסימום $maxsplit$ פעמים.

```
str.split("Hello my name is Amit Hi there!", ' ')
-> ['Hello', 'my', 'name', 'is', 'Amit', 'Hi', 'there!']

str.split("Hello my name is Amit Hi there!", ' ', 2)
-> ['Hello', 'my', 'name is Amit Hi there!']
```

ג

הפונקציה מקבלת מחרוזת עליה היא עובדת ועוד 2 מחרוזות old ו new ומחליפה כל מופע של old במחרוזת עליה היא עובדת בnew

```
str.replace("Hello Hello Hello".replace, ' ', ',')
-> "Hello,Hello,Hello"

str.replace("Hello! I don't like the word 'Hello'",
'Hello', 'Something')
-> "Something! I don't like the word 'Something'"
```

ד

הפונקציה מקבלת מחרוזת עליה היא עובדת ועוד רשימה של מחרוזות הפונקציה משרשרת את המחרוזות ברשימה יחד עם המחרוזת שהיא עובדת עליה ביניהן ומחזירה את התוצאה.

```
str.join(' ', ['hi', 'amit', 'banay'])
-> 'hi amit banay'

str.join("Hello", ["Hi = ", ', Hola = ', ' :)'])
-> 'Hi = Hello, Hola = Hello :)'
```

ה

הפונקציה מקבלת מחרוזת עליה היא עובדת ומחרוזת נוספת s הפונקציה מחזירה את מספר ההופעות של s במחרוזת עליה היא עובדת.

```
str.count('Hello sdf Hello fds Hello')
-> 3

str.count('abcda', 'a')
-> 2
```

ו

הפונקציה $find$ מקבלת מחרוזת עליה היא עובדת ומחרוזת נוספת s ומחזירה את האינדקס הנמוך ביותר שבו במחרוזת עליה היא עובדת מופיעה s. אם s לא נמצאת במחרוזת הפונקציה מחזירה -1. הפונקציה $index$ זהה אך במקום להחזיר ערך -1 כאשר s לא נמצאת היא זורקת $ValueError$.

```
str.find('hellop', 'lo')
-> 3

str.find('hellop', 'a')
-> -1

str.index('hellop', 'lo')
-> 3

str.index('hellop', 'a')
-> ValueError
```

ז

הפונקציה $append$ מקבלת רשימה עליה היא עובדת ואובייקט. הפונקציה מוסיפה את האוייקט לסוף הרשימה.

```
l = ['a']
list.append(l, 'b')
```

```
l == ['a', 'b']
-> True
```

הפונקציה $extend$ מקבלת רשימה עליה היא עובדת ורשימה נוספת l. הפונקציה מוסיפה לרשימה עליה היא עובדת את האיברים בl.

```
l1 = ['a']
l2 = ['a', 'b']
l1.extend(l2)
l1 == ['a', 'a', 'b']
-> True
```

הפונקציה pop מקבלת רשימה עליה היא עובדת ומספר שלם אי שלילי ומחזירה את האובייקט במיקום של האינדקס בתוך הרשימה.

```
l = ['a', 'b', 'c']
list.pop(l, 1)
-> 'b'
```

הפונקציה $remove$ מקבלת רשימה עליה היא עובדת ואובייקט ומוחקת את המופע הראשון שלו ברשימה. אם האובייקט לא נמצא ברשימה הפונקציה זורקת שגיאת $ValueError$

```
l = ['a', 'b', 'c']
list.remove(l, 'b')
l == ['a', 'c']
-> True
list.remove(l, 'b')
-> ValueError
```

הפונקציה $insert$ מקבלת רשימה עליה היא עובדת, אינדקס ואובייקט. הפונקציה מכניסה את האוייקט לתוך הרשימה לפני האיבר שנמצא באינדקס.

```
l = ['a', 'b', '1', '2']
list.insert(l, 2, 'c')
l == ['a', 'b', 'c', '1', '2']
True
```

הפונקציה $sort$ מקבלת רשימה עליה היא עובדת וממיינת אותה.

```
l = [1,5,2,4,3]
list.sort(l)
l == [1,2,3,4,5]
-> True
```

הפונקציה $count$ מקבלת רשימה עליה היא עובדת ואובייקט. הפונקציה מחזירה את כמות ההופעות של האוייקט ברשימה.

```
l = [1,2,3,3,4]
list.count(l, 1)
-> 1
list.count(l, 3)
-> 2
\end(lstlisting}

\selectlanguage{english}
\begin{lstlisting}
l = [1,2,3,3,4]
list.index(l, 3)
-> 2

list.index(l, 5)
-> ValueError
```

הפונקציה $index$ מקבלת רשימה עליה היא עובדת ואובייקט. הפונקציה מחזירה את האינדקס של האובייקט ברשימה עליה היא עובדת. אם האובייקט לא נמצא ברשימה הפונקציה זורקת $ValueError$.

שאלה ג

א

f1

power	time
200	2.2758601289751823e-05
400	5.399589713306341e-05
800	0.00016020270322769647
1600	0.0005551313727210072
3200	0.001958578453240989
6400	0.007663401179115681
12800	0.029707560898259544
25600	0.1169461884028351
51200	0.4659288119876237

f2

power	time
200	4.0162231016438454e-06
400	7.586199899378698e-06
800	1.204867112392094e-05
1600	2.0527365450107027e-05
3200	3.926974386558868e-05
6400	8.389445156353759e-05
12800	0.0002030424229815253
25600	0.0005631638205159106
51200	0.0018211343513030442

שתי הפונקציות גדלות בקצב שגדל ככל שהקלט גדל.

ב

f3

power	time
200	1.4726153779065498e-05
400	1.7849883363396657e-05
800	2.4097342532058974e-05
1600	3.748475506881732e-05
3200	9.638937021350102e-05
6400	0.00030389426441956857
12800	0.0011446237713244045
25600	0.004342430377647588
51200	0.017179620252676386

הפונקציה גדלה בקצב שגדל ככל שהקלט גדל. היא יעילה בהשוואה לפתרון הראשון אך לא בהשוואה לשני.

ג

f1 with random 1000 digits long numbers

number of zeros	time
0	0.0021879494552194956
52	0.002345028427953366
72	0.002077726424431603
81	0.0020937913195666624
90	0.0021053937434771797
100	0.002108517473061511
110	0.0021134261914994568
119	0.00218661071266979
128	0.0021839332302988623

ניתן לראות שמספר האפסים כמעט ואינו משפיע על זמן הריצה.

f2 with random 1000 digits long numbers

number of zeros	time
0	8.166321640601382e-05
70	8.25557108328212e-05
80	0.00010352932349633193
90	8.30019580462249e-05
100	8.30019580462249e-05
110	0.0001129005122493254
120	8.434069877694128e-05
130	8.434069877694128e-05

שוב ניתן לראות שמספר האפסים כמעט ואינו משפיע על זמן הריצה.

f3 with random 1000 digits long numbers

number of zeros	time
74	3.6592260585166514e-05
80	2.632857831486035e-05
90	2.722107274166774e-05
100	2.632857831486035e-05
110	2.6774825528264046e-05
120	2.7221071832173038e-05
131	2.7667319045576733e-05

שוב ניתן לראות שמספר האפסים כמעט ואינו משפיע על זמן הריצה. במקרה של 74 נראית עליה חריגה בזמן הריצה של הפונקציה.

ד

הלולאה תיקח המון זמן. הסיבה לכך היא שהפונקציות הקודמות רצות על כל הספרות של המספר שהן: $\lceil \log_{10} 2^{200} \rceil$ ואילו הלולאה השנייה רצה 2^{100} פעמים.

שאלה 6

$length = 20, start = 268382, seq =' 37579319339117379797'$