1. Dureaza mai mult sa adaugi elemente cu append intr-o lista sau sa folosesti list comprehension?

Creeaza 2 functii care returneaza timpul necesar celor 2 operatiuni.

Poti folosi time.time() pentru a obtine timpul curent.

1. Se da un string s. Scrie o functie care returneaza suma caracterelor de tip cifra din s.

Modifica functia astfel incat sa returneze suma numerelor din s. Un numar este reprezentat de caractere numerice alaturate.

ex:

s = 'eu am 33 de ani. in 2 luni va fi ziua mea de nume'

result: 35

1. Scrieti o functie care primeste un numar variabil de argumente de tip lista si returneaza o lista cu toate elementele argumentelor.
2. Scrieti o functie care primeste numele unui fisier ca parametreu si returneaza tipul acelui fisier.

ex:

def file\_type(file\_name):

pass

file\_type(‘20221107.jpeg’) -> imagine

file\_type(‘test.txt’’) -> text

file\_type(‘music\_20221107.mp3’) -> muzica

1. Scrieti o functie care determina daca un cuvant contine litere care nu se repeta (este format din caractere unice)

Functia poate returna True sau False

1. \* Se da un numar. Scrie o functie care returneaza numarul ca un string in urmatorul format:

Input: 37 → output: ‘30 + 7’

Input: 114 → output: ‘100 + 10 + 4’

Input: 30165 → output: ‘30000 + 100 + 60 + 5’

1. Scrie o functie care returneaza cuvantul cu scorul cel mai mare.

Scorul unui cuvant este dat de suma pozitiilor literelor cuvantului, din alfabet.

Functia primeste ca argument o lista de cuvinte.

ex:

a = 1, b = 2, c = 3, d = 4, etc

Word: mama, scor: 13 + 1 + 13 + 1 = 30

1. Se dau 3 tupluri ca input. Fiecare dintre aceste tupluri contine urmatoarele informatii (nume, varsta, inaltime) Afiseaza cele 3 tupluri, dupa sortarea (a-z) si in ordine crescatoare, tinand cont, pe rand, de nume, varsta si inaltime.   
   Ex.

input: (Dan, 33, 170), (Mihai, 20, 180), (Daniel, 40, 173)

dupa nume: (Dan, 33, 170), (Daniel, 40, 173), (Mihai, 20, 180)

dupa varsta: (Mihai, 20, 180), (Dan, 33, 170), (Daniel, 40, 173)

dupa inaltime: (Dan, 33, 170), (Daniel, 40, 173), (Mihai, 20, 180)

1. Se da o lista de intregi. Pentru fiecare element din lista sa se determine cate numere din cadrul acelei liste sunt mai mici decat acel element. Rezultatul se returneaza tot intr-o lista

Ex. nbs = [3, 7, 8, 5]

output: [0, 2, 3, 1]

pentru nbs[0] = 3 nu exista niciun numar mai mic in nbs pentru nbs[1] = 7 exista 2 numere mai mici, etc.

1. \* Fie un intreg pozitiv n, dat ca input. Scrie o aplicatie care sa printeze o forma triunghiulara realizata din caracterul '\*'.

n reprezinta numarul de randuri ale acestui pattern. Numarul de pe fiecare rand va creste cu 2.

Ex. input n = 3

output:

\*

\*\*\*

\*\*\*\*\*

input: n = 5

output:

\*

\*\*\*

\*\*\*\*\*

\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*

1. Fie un string s care contine cuvinte separate de unul sau mai multe spatii. Sa se returneaza un string cu toate cuvintele inversate si separate de un singur spatiu.

Ex. input s = 'python java javascript'

output: 'javascript java python'