

1. Se citește un vector de numere naturale (cu elementele date pe o linie, separate prin spațiu). Să de ordoneze elementele din vector crescător după suma cifrelor, iar în caz de egalitate, descrescător după valorile lor

$v = [11, 45, 20, 810, 179, 81, 1000] \Rightarrow v = [1000, 20, 11, 810, 81, 45, 179]$

2. a) Scrieți o funcție care primește un număr variabil de parametri și un filtru = o funcție booleană și returnează o listă cu parametri care verifică filtrul. Dacă la apel nu este transmisă nicio funcție pentru filtru, atunci se vor returna parametrii primiți.

Exemplu de apel:

```
def pozitiv(x):  
    return x>0  
a = filtreaza(3,-1,6,8,-3,functie=pozitiv)  
print(a)  
a = filtreaza(3,-1,6,8,-3)  
print(a)  
a = filtreaza("ana","are","10","mere",functie=str.isalpha)  
print(a)
```

b) Modificați antetul de la a astfel încât funcția să primească parametru o listă pe care să o filtreze (nu un număr variabil de parametri). Exemplu de apel:

```
a = filtreaza([3,-1,6,8,-3],functie=pozitiv)  
print(a)
```

3. a) Scrieți o funcție care primește un număr întreg x și un număr variabil de liste nevide de numere întregi și returnează numărul de liste primite ca parametru care conțin x.

Exemplu apel:

```
nr = liste_x(3, [1, 5, 7], [3], [1, 8, 3], [])  
print(nr)
```

b) Modificați funcția de la a astfel încât rezultatul să nu fie returnat, ci să se salveze în variabila globală **rez**. Exemplu apel:

```
rez = None  
liste_x(3, [1, 5, 7], [3], [1, 8, 3], [4,3])  
print(rez)
```

4. a) Scrieți o funcție generică de căutare având următorul antet: **cautare(x, L, cmpValori)** Funcția trebuie să returneze indexul ultimei apariții a valorii x în lista L sau None dacă valoarea x nu se găsește în listă. Funcția comparator cmpValori se consideră că primește 2 parametri și returnează True dacă valorile primite ca parametrii sunt egale sau False în caz contrar.

b) Se consideră lista de perechi (tupluri) l_pairs ale cărei elemente se citesc din fișierul perechi.txt (numerele dintr-o pereche sunt date pe o linie, separate prin -). Se citesc două numere x și y de la tastatură. Folosind un singur apel al funcției de la a) să se verifice dacă lista conține perechea (x,y) sau perechea (y,x) și, în caz afirmativ, afișează indexul ultimei apariții ale unei astfel de perechi.

c) (Temă) Scrieți o funcție care să afișeze, folosind apeluri utile ale funcției cautare, mesajul DA în cazul în care o listă L formată din n numere întregi este modulo-palindrom sau mesajul NU în caz contrar. O listă este modulo-palindrom dacă prin parcurgerea modulelor elementelor sale de la dreapta la stânga sau de la stânga la dreapta se obține aceeași listă.

De exemplu, lista $L=[101,17,-101,13,5,-13,101,17,-101]$ este palindrom.

5. Care este complexitatea următorului algoritm?

a)

```
s = 0
i = n
while i >= 1:
    j = 1
    while j <= n:
        s = s + 1
        j = j + 1
    i = i/2
```

b)

```
s = 0
i = n
while i >= 1:
    j = 1
    while j <= i:
        s = s + 1
        j = j + 1
    i = i/2
```

Temă: a) 2-SUM Se dă un vector cu n elemente reale/întregi distincte. Să se afișeze toate perechile de elemente din vector cu suma 0. Care este complexitatea algoritmului propus?

b) 3-SUM Se dă un vector cu n elemente reale/întregi distincte. Să se afișeze toate tripletele de elemente din vector cu suma 0. Care este complexitatea algoritmului propus?

Metoda Greedy - lista probleme (seminar + laborator)