

1. a) Scrieți o funcție care primește 2 parametri m, n și returnează o matrice mxn cu elemente numere naturale <100 generate aleator.

b) Scrieți o funcție care afișează o matrice **globală** numita a

c) Scrieți o funcție care primește ca parametru o matrice și o **modifica** ordonând-o descrescător după ultima coloana (prin interschimbare de linii)

Exemplu apeluri:

```
a = gen_matrice(4,3)
print_matrice()
ordoneaza(a)
print_matrice()
```

2. 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)
```

c) Modificați funcția de la a astfel încât sa returneze primele 3 liste primite ca parametru care conțin x (sau liste vide dacă nu există). Exemplu apel:

```
t = liste_x(3, [1,5,7], [3], [1,8,3], [2])
print(t)
x,y,z = liste_x(3, [1,5,7], [3], [1,8,3], [2,3], [4])
print(x,y)
```

3. Se dă un dicționar d care conține informații de forma

```
cod_student=[grupa,nume,lista_note]
```

(cheia este un cod asociat unui student, iar valoarea este o lista conținând următoarele

informații despre student: grupa, nume, lista note). Un exemplu de astfel de dicționar este:

```
d={1: [242, "Ana", [10,4,7]], 2: [241, "Ioana", [10,10,9]], 3: [241, "Ion", [10,10,7]]
, 4: [242, "Matei", [10,5,9]]}
```

a) Scrieți o funcție care primește ca parametru un dicționar de forma celui descris și returnează o listă cu informațiile despre studenți cu elemente de forma

```
[grupa, nume, lista_note]
```

ordonata după grupă, în caz de egalitate descrescător după media notelor, iar în caz de egalitate după nume. Exemplu de apel:

```
print(lista_grupe(d))
```

b) Scrieți o funcție care primește ca parametru un dicționar de forma celui descris și adaugă la începutul listei de informații asociate unui student un cod de lungime generat aleator care conține 3 litere urmate de 3 cifre. Exemplu de apel:

```
genereaza_coduri(d)
print(d)
```

4. a) Scrieți o funcție care primește ca parametru un număr n și o funcție f și calculează suma $\sum_{i=1}^n f(i)$. Dacă f nu este specificat se va calcula suma $\sum_{i=1}^n i$.

Parametrul f va fi (obligatoriu) specificat prin nume

Exemplu apel:

```
print(suma(10))
import math
print(suma(10,f=math.sqrt))
def radical(x):
    return x**0.5
print(suma(10,f=radical))
```

b) Scrieți o funcție care primește ca parametri un număr variabil de numere x_1, \dots, x_n și o funcție f și calculează suma $\sum_{i=1}^n f(x_i)$. Dacă f nu este specificat se va calcula suma $\sum_{i=1}^n x_i$.

Parametrul f va fi obligatoriu specificat prin nume

Exemplu apel:

```
print(suma(2,3,4,5))
ls = [3,1,7]
print(suma(*ls))
#print(suma(ls))
from math import sqrt
print(suma(4,9,16,f=sqrt))
```

5. Scrieți o funcție care primește un număr variabil de parametri și un filtru = o funcție booleană și returnează parametri care verifică filtrul:

a) ca lista

b) ca generator

Exemplu de apel:

```
def pozitiv(x): return x>0
```

a) ca o noua lista

```
a = filtreaza(3,-1,6,8,-3,functie=pozitiv)
print(a)
a = filtreaza("ana","are","10","mere",functie=str.isalpha)
print(a)
```

b) un generator

```
a = filtreaza(3,-1,6,8,-3,functie=pozitiv)
print(a)
print(*a)
print(sum(a))
```

c) 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)
```

6. Exemplu generator

Temă

1. a) Scrieți o funcție cu număr variabil de parametri care să furnizeze numărul natural obținut prin alipirea cifrelor maxime ale numerelor naturale nenule primite ca parametri. De exemplu, pentru numerele 4251, 73, 8 și 133 funcția trebuie să returneze numărul 5783.
b) Scrieți o funcție cu 3 parametri nenuli de tip întreg a,b și c care să verifice dacă aceștia pot fi considerați ca fiind numere scrise în baza 2 sau nu, folosind apeluri utile ale funcției definite anterior. De exemplu, pentru numerele 1001, 11 și 100 funcția trebuie să returneze valoarea True, iar pentru numerele 1001, 17 și 100 trebuie să returneze valoarea False.
2. Scrieți o funcție cu număr variabil de parametri care să caute un cuvânt dat în mai multe fișiere text. Funcția va scrie într-un fișier text câte o linie pentru fiecare fișier text de intrare, astfel: numele fișierului text de intrare și apoi numerele de ordine ale liniilor pe care apare cuvântul dat în acel fișier sau un mesaj corespunzător dacă fișierul nu conține cuvântul respectiv.
3. Fișierul text “angajati.txt” conține pe fiecare linie numele, vârsta și salariul câte unui angajat, despărțite prin virgule. Scrieți un program care să încarce datele din fișierul text în memorie și să afișeze următoarele informații:
 - a) informațiile despre un angajat al cărui nume se citește de la tastatură (pe ecran);
 - b) salariul maxim și numele angajaților care au salariul respectiv (pe ecran);
 - c) salariul mediu din firmă (pe ecran);
 - d) angajații sortați alfabetic după nume (în fișierul text “angajati_nume.txt”);
 - e) angajații sortați descrescător după vârstă și apoi alfabetic după nume (în fișierul text “angajati_varsta_nume.txt”);
 - f) angajații sortați descrescător după salariu, apoi crescător după vârstă (în fișierul text “angajati_salariu_varsta.txt”).

Fiecare cerință se va rezolva în cadrul unei funcții!