

Programare Procedurală

- Laborator 4 -

Observații:

- **Toate datele** de intrare pentru care testați programele scrise se vor introduce tot pe câte un rand (nu introducem mai multe numere pe aceeași linie).
- Evitați pe cât este posibil folosirea structurilor repetitive și faceți uz de **metodele** unui obiect.

1) Care este outputul următoarelor comenzi:

- a) `print([n ** 2 for n in (0, 1, 2, 3)])`
- b) `print((n ** 2 for n in (0, 1, 2, 3) if n > 0))`
- c) `print({n ** 2 for n in range(-3, 3)})`

Răspundeți, justificați și apoi rulați pentru a vă verifica răspunsul.

2) Care este outputul următoarelor comenzi:

- a) `print({1, 2, 3} * 2)`
- b) `print({1, 2, 3} * 0)`
- c) `print({1, 2, 3} * -2)`
- d) `print({1, 2, 3} + {4, 5, 6})`
- e) `print({1, 2, 3} + {0, 1, 2})`

Răspundeți, justificați și apoi rulați pentru a vă verifica răspunsul.

3) Care este diferența dintre copierea simplă și copierea profundă a mulțimilor? Dar pentru tuple? Argumentați.

4) Se citește n , apoi un șir de n numere întregi. Să se afișeze:

- a. câte perechi de numere identice se pot forma extragand numere din șir.
Explicație: pentru $n = 11$ și șirul 3, 6, 4, 3, 2, 6, 3, 4, 5, 3, 6 se pot forma 4 perechi: (3,3), (3,3), (4,4), (6,6).

INPUT:

11
3
6
4
3
2
6
3
4
5
3
6

OUTPUT:

4

- b. Modificați cerința a) pentru a afișa perechile.

Exemplu:

INPUT

11
3
6

4
3
2
6
3
4
5
3
6
OUTPUT
(3,3), (3,3), (4,4), (6,6)

5) Institutia “Carpe diem” are n angajati (n citit de la tastatura). Institutia are program de lucru intre orele a , b (citite de la tastatura). Pentru fiecare angajat se citeste (de la tastatura) ora la care are o activitate in institutie. Orice activitatea dureaza exact o ora. Seful doreste sa organizeze o sedinta (de o ora) in timpul programului de lucru, care sa nu se suprapuna cu nicio activitate din institutie (astfel incat sa poata participa toti angajatii). Care sunt orele disponibile (daca exista) dintre care seful poate alege sa organizeze sedinta?

Exemplu:

INPUT
n = 7 (angajati)
a = 9
b = 17
Orele la care incep activitatile:
12
10
14
14
10
15
9
OUTPUT
Seful poate alege dintre orele: 11, 13, 16.

- 6) Se citește n , apoi două șiruri având câte n numere întregi (valorile se pot repeta). Să se afișeze:
- valorile comune celor două șiruri;
 - valorile comune celor două șiruri (cu tot cu repetiții);
Exemplu: pentru $n = 8$ și șirurile $s1$: 1, 2, 2, 3, 4, 4, 4, 5 și $s2$: 2, 2, 2, 3, 4, 4, 5, 5 se va afișa: 2, 2, 3, 4, 4, 5;
 - valorile care apar în plus în primul șir față de al doilea (se iau în considerare repetițiile)
Exemplu: pentru $n = 8$ și șirurile $s1$: 1, 2, 2, 2, 1, 4, 4, 4 și $s2$: 2, 3, 4, 4, 5, 5, 7, 8 se va afișa: 1, 1, 2, 2, 4.
- 7) Sa se citeasca un numar natural n , apoi n valori numere intregi ce se vor memora intr-o lista l . Scrieți un program care să extragă duplicatele din lista l într-o alta, numită *dubluri*, lasand lista initiala cu valori unice. Nu ne intereseaza ordinea elementelor din liste.

Exemplu:

Daca dupa citirea datelor lista memorata este $l = [2, 3, 2, 2, 2, 4, 4]$, se va crea lista *dubluri* = $[2, 2, 2, 4]$, iar continutul listei l va fi $l = [2, 3, 4]$.

- 8) Se citește un șir s de la tastatură. Să se afișeze literele mici ale alfabetului englez care nu apar în șirul s .

Exemplu:

INPUT

Cateva cuvinte mostenite probabil din limba dacă: abur, aidoma, amurg, aprig, balta, brusture, codru, copac, descurca, gutui, mazare etc.

OUTPUT

f h j k q w x y

Observație:

Putem folosi:

- `ord(c)`: codul ASCII al caracterului c ;
- `chr(x)`: caracterul cu codul ASCII x .

- 9) Scrieți un program care:

- a) realizează un tabel de frecvențe pentru literele mici dintr-un text introdus de la tastatură.

Exemplu:

INPUT

ana are mere

OUTPUT

e : 3

n : 1

a : 3

m : 1

r : 2

- b) rezolvă cerința a) cu tabelul ordonat descrescător după numărul de apariții (nu este necesar parametrul *key* al metodei *sort* – vom detalia în cursurile următoare, la subiectul *Functii*);

Exemplu:

INPUT

ana are mere

OUTPUT

e : 3

a : 3

r : 2

n : 1

m : 1