

Test laborator

Varianta 2

1.

- a) Se citesc două şiruri de numere, de la tastatură, de pe o singură linie, separate prin câte un spaţiu. Să se memoreze cele două şiruri de valori. **(1p.)**
- b) Să se afișeze valorile comune celor două şiruri, o singură dată, ordonate descrescător (se va folosi funcția *sorted*). **(1p.)**
- c) Să se creeze utilizând comprehensiunea un tuplu T ce va conține doar numerele pozitive comune celor două şiruri. Să se afișeze tuplul obținut. **(1p.)**
- d) Să se folosească funcția *map* pentru a crea o listă ce conține primele două cifre (sau cifra, pentru numerele mai mici decât 10), din fiecare număr din tuplul T. **(1p.)**

Exemplu

INPUT

consola

```
12 -1 345 -1 2 3 -1 12 345 5  
-1 13 14 -1 -1 17 345 2
```

OUTPUT

consola

- b) 345 2 -1
- c) (2, 345)
- d) [2, 34]

2. Fie fișierul *date2.txt* (atașat), ce conține, pe câte o linie informații despre produsele unui magazin: cod, categorie, subcategorie, nume, preț, eventuale observații.

- a. Să se memoreze produsele într-un dicționar ce are drept chei categoriile, iar valoarea asociată fiecarei chei este un dicționar ce are drept chei subcategoriile produselor. Valoarea asociată fiecărei chei ce identifică o subcategorie trebuie să conțină toate produsele respectivei subcategorii, împreună cu toate informațiile (cod, nume, preț, observații). Afipați dicționarul obținut. **(1.5p.)**
- b. Să se afișeze în consolă fiecare categorie, în cadrul căruia este afișată fiecare subcategorie, împreună cu toate informațiile produsului celui mai ieftin. **(1p.)**
- c. Scrieți o funcție numită *creare_fisiere* ce primește ca argumente:
 - o listă variabilă de şiruri de caractere, ce reprezintă nume de subcategorii de produse;
 - un dicționar cu aceeași structură ca cel creat la cerința 2.a).

Funcția creează câte un fișier cu numele fiecărei subcategorii din cele date ca argumente și extensia *.txt* (*lactate.txt*, *dulciuri.txt*, etc...) ce conține, pe câte o linie, codul, numele, prețul și observațiile pentru fiecare produs din subcategoria respectivă. Să se apeleze funcția. **(1.5p.)**

- d. Scrieți o funcție numită *marca* care primește ca argument numele unui fișier și, printr-o singură linie de cod, folosind expresii regulate, fără a utiliza structuri repetitive, returnează lista tuturor mărcilor din fișierul dat ca argument (mărcile sunt cuvinte care încep cu majuscule). Să se apeleze funcția pentru unul dintre fișierele create prin rezolvarea cerinței 2.c). **(1p.)**