

1.

a) [0,5p] Scrieți o funcție **citire_matrice** cu un parametru reprezentând numele unui fișier care conține elementele unei matrice de numere naturale cu următoarea structură: pe linia *i* a fișierului sunt elementele de pe linia *i* a matricei separate printr-un spațiu (vezi exemplul de fișier de intrare la punctul c)). Funcția citește elementele matricei din fișierul cu numele dat ca parametru și returnează matricea cu aceste elemente. Dacă în fișierul de intrare numărul de numere de pe fiecare linie nu este același pentru toate liniile funcția va returna **None**.

b) [1,25p] Scrieți o funcție **multimi** care primește ca parametri (în această ordine): o matrice și un număr variabil de numere naturale reprezentând indici ai liniilor din matrice (indicele primei linii din matrice este 0; indicii dați sunt mai mici decât numărul de linii ale matricei). Asociem fiecărei linii din matrice două mulțimi: mulțimea elementelor negative și mulțimea elementelor pozitive care au prima cifră egală cu ultima.

Funcția returnează următoarele două mulțimi:

- intersecția mulțimilor elementelor negative asociate liniilor corespunzătoare indicilor dați
- reuniunea mulțimilor elementelor pozitive care au prima cifră egală cu ultima asociate liniilor corespunzătoare indicilor dați

(elementele din reuniune sunt distincte două câte două, la fel și cele din intersecție).

Se acordă jumătate din punctaj dacă în loc de o funcție cu număr variabil de parametri se va scrie o funcție **multimi** care primește 2 parametri (în această ordine): o matrice și o listă de numere naturale reprezentând indici ai liniilor din matrice și returnează informațiile cerute la punctul b).

c) [1,25p] Se dă fișierul "**matrice.in**" cu structura descrisă la punctul a). Folosind apeluri utile ale funcțiilor de la a) și b) să se citească matricea din fișierul "**matrice.in**" și să se afișeze pe ecran numerele pozitive cu prima cifră egală cu ultima care se află în fișier pe ultimele 3 linii (se vor afișa pe aceeași linie, separate prin spațiu, ordonate crescător), precum și numărul de elemente negative care se află atât pe prima cât și pe ultima linie din fișier.

Pentru punctul c) se acordă **1p** dacă este rezolvat corect dar fără a folosi funcția de la b).

Exemplu:

matrice.in	Iesire pe ecran
-3 10 -1 -3 10 1 1 14 1 212 45 73 131 -3 122 1131	1 131 212 1131 1

Explicații – numărul -3 este singurul număr negativ care se află și pe prima și pe ultima linie.

2. a) [1p] Scrieți o funcție **modifica_prefix** cu 3 parametri șiruri de caractere: **x**, **y**, **prop** (în această ordine), unde **x** și **y** sunt două șiruri diferite de caractere formate doar din litere, iar **prop** este o propoziție în care cuvintele sunt separate prin câte un spațiu. Funcția returnează două valori:

- propoziția obținută modificând propoziția **prop** astfel: în fiecare cuvânt care începe cu **x** prefixul **x** este înlocuit cu **y**
- numărul de cuvinte al căror prefix a fost modificat

b) [1p] Scrieți o funcție **poz_max** cu un parametru, care primește ca parametru o listă de numere naturale și returnează pozițiile pe care apare maximul în listă (numerotate de la 1)

c) [1p] Se dă fișierul "**propozitii.in**" cu următoarea structură:

- pe linia **k** a fișierului se află o propoziție cu cuvintele separate prin câte un spațiu; spunem că propoziția de pe linia **k** are indicele **k** (cu numerotarea începând de la 1).

Se citesc de la tastatură două cuvinte diferite **a** și **b** (formate doar din litere; cele două cuvinte se dau pe o linie, separate prin spațiu). Folosind apeluri utile ale funcțiilor de la a) și b) să se rezolve următoarele cerințe:

- să se creeze un nou fișier "**propozitii_modificate.out**" cu propozițiile din fișierul "**propozitii.in**" modificate astfel: în fiecare cuvânt care începe cu **a** prefixul **a** este înlocuit cu **b**

- să se afișeze pe ecran indicii **k** ai propozițiilor în care s-au făcut cele mai multe modificări (cu numerotarea începând de la 1), separați prin câte un spațiu.

Pentru punctul c) se acorda **0,75p** dacă este rezolvat corect dar fără a folosi funcțiile de la a) și b).

Exemplu:

propozitii.in	propozitii_modificate.out
aceasta este ceasa in ceare locuiesc	aceasta este casa in care locuiesc
parca ar fi o ceasuta din povesti	parca ar fi o casuta din povesti
este plina de ceadouri si cearti	este plina de cadouri si carti
sunt si multe ceaiete si creioane	sunt si multe caiete si creioane
este cea mai frumoasa casa	este ca mai frumoasa casa
Intrare de la tastatura	Iesire pe ecran
cea ca	1 3

3. Se dă fișierul “**autori.in**” cu următoarea structură:

- Pe prima linie sunt două numere naturale **m** și **n** separate printr-un spațiu.
- Pe următoarele **m** linii sunt câte 3 valori separate prin spațiu reprezentând informații despre un autor: codul (număr natural), numele și prenumele unui autor.
- Pe următoarele **n** linii sunt valori separate prin spațiu reprezentând 5 informații despre cărți scrise de autorii dați anterior (o carte are un unic autor): codul unui autor (număr natural, dintre codurile date pe liniile 2, ..., m+1), codul cărții (număr natural), an apariție, număr de pagini, numele cărții (șir ce poate conține spații).

```
autori.in
3 7
11 Ionescu Ion
20 Popescu Marin
17 Georgescu Oana
11 131 2010 30 Despre algoritmi
11 101 1990 70 Bazele Informaticii
20 213 2012 80 Introducere in C/C++
11 141 2010 100 Complexitatea algoritmilor
11 121 2010 100 Tehnici de programare
17 214 2018 150 Introducere in Python
20 111 2017 50 Initiere in programare
```

a. **[1,25p]** Să se memoreze datele din fișier într-o singură structură astfel încât să se răspundă cât mai eficient la cerințele *b*) (ștergerea unei cărți având dat codul cărții și aflarea numelui unicului său autor) și *c*) (accesarea numelui unui autor și a informațiilor despre toate cărțile sale, având dat codul autorului).

b. **[0,75p]** Să se scrie o funcție **sterge_carte** cu 2 parametri: în primul parametru se transmite structura în care s-au memorat datele la cerința a), iar al doilea este codul unei cărți, care șterge din structura de date primită toate informațiile legate de cartea cu codul dat ca parametru. Funcția returnează numele unicului autor al cărții cu codul dat, sau **None** dacă acea carte nu s-a găsit.

Să se apeleze funcția pentru un cod de carte citit de la tastatură și să se afișeze pe ecran mesajul “Cartea a fost scrisa de”, sau mesajul “Cartea nu exista.”. Apoi să se afișeze pe ecran toată structura rămasă după ștergere.

Intrare de la tastatură	Ieșire pe ecran
111	Cartea a fost scrisa de Popescu Marin. ...structura de date afisata
333	Cartea nu exista.

c. **[1p]** Să se scrie o funcție **carti_autor** cu 2 parametri: în primul parametru se transmite structura în care s-au memorat datele la cerința a), iar al doilea este codul unui autor. Funcția returnează numele autorului și o listă cu informații despre cărțile sale (un element al listei fiind un tuplu ce conține: numele cărții, anul apariției, numărul de pagini), lista fiind sortată crescător după anul apariției, în caz de egalitate descrescător după numărul de pagini, iar în caz de egalitate crescător după numele cărții. Funcția va returna o listă vidă dacă nu există un autor cu codul primit ca parametru.

Să se apeleze funcția pentru un cod de autor citit de la tastatură și să se afișeze rezultatul returnat ca în exemplul de mai jos.

Intrare de la tastatură	Ieșire pe ecran
11	Ionescu Ion Bazele Informaticii 1990 70 Complexitatea algoritmilor 2010 100 Tehnici de programare 2010 100 Despre algoritmi 2010 30
10	cod incorect