TEST DE LABORATOR LA DISCIPLINA "PROGRAMAREA ALGORITMILOR"
VARIANTA 3

Subiectul 1 – 4 p.

a) [0,5 p.] Considerăm un fișier text având următoarea structură: pe prima linie se află un număr natural $n \geq 1$, iar pe fiecare din următoarele linii sunt scrise câte două numere naturale x și y (numărul y este cuprins între 0 și $n-1$), despărțite între ele prin câte un spațiu. Scrieți o funcție ***citire_liste*** cu un parametru reprezentând numele unui fișier text având structura precizată anterior care returnează o listă formată din n liste (numite subliste), unde sublista cu indexul $0 \leq k \leq n-1$ va conține toate valorile x din perechile de forma $x\ \ k$. Pentru fișierul din exemplu matricea returnată va fi $[[56, 21, 8, 0, 74], [], [8, 10, 0, 8, 0], [], [12, 21], [100, 0]]$.

b) [2 p.] Să se scrie o funcție ***prelucrare_liste*** care primește ca parametru o listă de liste L și un număr întreg x și modifică lista L astfel:

- din fiecare sublistă din lista L se vor elimina toate aparițiile valorii x, apoi
- se vor șterge din L toate sublistele care conțin cel mult un element.

c) [0,5 p.] Se dă fișierul ***liste.in*** cu o structură de tipul celei precizate anterior. Să se apeleze funcția *prelucrare_liste* pentru lista de liste L obținută în urma apelului funcției *citire_liste* pentru fișierul text *liste.in* și parametrul x egal cu 0. Lista de liste astfel obținută să se afișeze pe ecran, fără paranteze și virgule, fiecare sublistă pe câte o linie, iar elementele unei sublistei să fie separate între ele prin câte un spațiu.

d) [1 p.] Fie lista de liste L obținută în urma apelării funcției *citire_liste* pentru fișierul text *liste.in*. Să se citească de la tastatură un număr natural nenul k și apoi să se scrie în fișierul text *divizori.out* elementele din sublistele listei L care au exact k divizori sau mesajul "*Imposibil!*" dacă nu există niciun element cu proprietatea cerută. Elementele vor fi scrise în fișier în ordine descrescătoare, câte unul pe linie și fără duplicate .

Exemplu:

liste.in	Punctul c) ecran	Punctul d) fișierul divizori.out pentru k = 4
6 8 2 10 2 12 4 56 0 0 2 21 0 8 2 21 4 0 2 8 0 100 5 0 0 0 5 74 0	56 21 8 74 8 10 8 12 21	74 21 10 8

Subiectul 2 – 5 p.

Moș Crăciun le-a cerut spiridușilor în luna noiembrie să scrie într-un fișier ***spiridusi.in*** ce jucării pot face până la Crăciun și câte bucăți. Un spiriduș poate scrie mai multe linii în fișier, chiar și cu aceeași jucărie, dacă se hotărăște că poate face mai multe bucăți. O linie din acest fișier are următoarea structură:

nume_spiridus : nume_jucarie nr_bucati

unde *nume_spiridus* este numele spiridușului, *nume_jucarie* este numele jucăriei, iar *nr_bucati* este numărul de bucăți (cantitatea) pe care spiridușul o poate produce din această jucărie; numele spiridușului și numele jucăriei pot conține litere și spații.

Un exemplu de fișier este:

```
Spiridus Poznas : papusa 1
Spiridus Jucaus : papusa 1
Spiridus Harnic : masinuta 1
Spiridus Poznas : trenulet electric 10
Spiridus Jucaus : papusa 1
Spiridus Jucaus : masinuta 2
Spiridus Poznas : ponei 11
Spiridus Harnic : ponei 15
Spiridus Poznas : papusa 2
Spiridus Harnic : papusa 3
Alt Spiridus : ursulet plus 11
```

(***Explicație:*** pentru datele din fișier *Spiridus Poznas* poate produce 1 + 2 = 3 păpuși)

a) [2,5 p.] Să se memoreze datele din fișier într-o singură structură de date astfel încât să se răspundă cât mai eficient la cerințele de la punctele următoare.

b) [1,5 p.] Scrieți o funcție ***top_spiridusi*** care primește următorii parametri (în această ordine):

- structura în care s-au memorat datele la cerința a),
- un număr variabil de șiruri de caractere reprezentând nume de spiriduși
- un număr *nr_minim*

Funcția returnează o listă de tupluri de forma (*nume_spiridus*, *multime_jucarii*, *cantitate_totala*) cu informații despre jucăriile din care spiridușii dați ca parametru produc minim *nr_minim* bucăți, unde:

- *nume_spiridus* este numele unui astfel de spiriduș
- *multime_jucarii* este mulțimea (nevidă a) jucăriilor din care spiridușul poate produce cel puțin *nr_minim* bucăți
- *cantitate_totala* este numărul total de bucăți pe care spiridușul le poate produce din jucăriile din mulțimea *multime_jucarii.*

Lista returnată va fi ordonată descrescător după numărul de elemente din *multime_jucarii* și, în caz de egalitate, descrescător după cantitatea totală de jucării *cantitate_totala* și, în caz de egalitate, crescător după numele spiridușului.

De exemplu, pentru datele din fișier și spiridușii "*Spiridus Harnic*", "*Spiridus Poznas*", "*Spiridus Jucaus*", dacă *nr_minim*=3 rezultatul returnat va *fi*
[('Spiridus Poznas', {'ponei', 'trenulet electric', 'papusa'}, 24), ('Spiridus Harnic', {'ponei', 'papusa'}, 18)]
("*Spiridus Jucaus*" produce 2 păpuși și 2 mașinuțe, deci pentru nicio jucărie nu produce minim *nr_minim* bucăți, de aceea el nu apare în lista rezultat),
iar pentru *nr_minim*=2 rezultatul va fi
[('Spiridus Poznas', {'ponei', 'trenulet electric', 'papusa'}, 24), ('Spiridus Harnic', {'ponei', 'papusa'}, 18), ('Spiridus Jucaus', {'masinuta', 'papusa'}, 4)].

c) [1 p.] Scrieți o funcție ***adauga_bucati*** care are următorii parametri (în această ordine):

- structura în care s-au memorat datele la cerința a)
- un șir de caractere *nume_spiridus* reprezentând numele unui spiriduș
- un șir de caractere *nume_jucarie* reprezentând numele unei jucării, care are valoare implicită șirul vid
- un parametru *nr_bucati* care are valoarea implicită 1

- Dacă numele jucăriei este un șir nevid, funcția va crește cu *nr_bucati* numărul de bucăți pe care le poate face spiridușul *nume_spiridus* din jucaria *nume_jucarie* (dacă spiridușul nu producea încă această jucărie, atunci cantitatea produsă după actualizare va fi *nr_bucati*).
- Dacă numele jucăriei este șirul vid, funcția va crește cu *nr_bucati* numărul de bucăți pe care le poate face spiridușul *nume_spiridus* pentru fiecare jucărie pe care acesta o produce deja.
Funcția va returna cantitatea totală de jucării produse de spiridușul *nume_spiridus* (funcția va returna 0 dacă spiridușul nu produce jucării).

Se citesc de la tastatură un nume de spiriduș *s* și un nume de jucărie *j*, date fiecare pe câte o linie. Să se apeleze funcția *adauga_bucati* pentru a crește cu 1 numărul de bucăți din jucăria *j* produse de spiridușul *s* (la apel nu se va transmite valoare parametrului *nr_bucati*, se va folosi valoarea implicită) și să se afișeze cantitatea returnată; după apelul funcției să se afișeze și structura în care s-au memorat datele.

MODALITATEA DE DESFĂȘURARE A TESTULUI DE LABORATOR

- Testul de laborator la disciplina "Programarea algoritmilor" se va desfășura în ziua de **15.01.2023**, în două runde, între orele 9^{00} și 11^{00}, respectiv 11^{30} și 13^{30}, astfel:
  - **Prima rundă**
    - **09^{00} – 09^{15}**: efectuarea prezenței studenților
    - **09^{15} – 10^{45}**: desfășurarea testului
    - **10^{45} – 11^{00}**: trimiterea surselor folosind un formular Google dedicat
  - **A doua rundă**
    - **11^{30} – 11^{45}**: efectuarea prezenței studenților
    - **11^{45} – 13^{15}**: desfășurarea testului
    - **13^{15} – 13^{30}**: trimiterea surselor folosind un formular Google dedicat

- Testul de laborator se va desfășura în laboratoarele Facultății de Matematică și Informatică, folosind calculatoarele din ele.
- Calculatoarele din laboratoare vor conține documentația oficială a limbajului Python, în format offline.
- Pe parcursul testului este interzisă utilizarea Internet-ului sau a oricărei alte forme de comunicare/informare, cu excepția documentației offline.

- În momentul efectuării prezenței, fiecare student trebuie să prezinte buletinul sau cartea de identitate.

- Testul va conține **două subiecte**, iar un subiect poate să aibă mai multe cerințe.

- Rezolvarea unui subiect se va realiza într-un singur fișier sursă Python (.py), indiferent de numărul de cerințe, care va fi încărcat/atașat ca răspuns pentru subiectul respectiv.

- Numele fișierului sursă Python trebuie să respecte următorul șablon: ***grupa_nume_prenume_subiect.py***. De exemplu, un student cu numele Popescu Ion Mihai din grupa 131 trebuie să denumească fișierul care conține rezolvarea primului subiect astfel: **131_Popescu_Ion_Mihai_1.py.**

- La începutul fiecărui fișier sursă Python se vor scrie, sub forma unor comentarii, numele complet al studentului și grupa sa. Dacă un student nu reușește să rezolve deloc un anumit subiect, totuși el va trebui să încarce/atașeze un fișier sursă Python cu informațiile menționate anterior!

- Toate rezolvările (fișierele sursă Python) trimise de către studenți vor fi verificate din punct de vedere al similarității folosind un software specializat, iar eventualele fraude vor fi sancționate conform Regulamentului de etică și profesionalism al FMI (http://old.fmi.unibuc.ro/ro/pdf/2015/consiliu/Regulament_etica_FMI.pdf).