

**TEST DE LABORATOR LA DISCIPLINA "PROGRAMAREA ALGORITMIILOR"**  
**VARIANTA 3**

**Subiectul 1 – 4 p**

Fișierul **"numere.in"** are  $n > 1$  linii care conțin fiecare, separate prin câte un spațiu, câte  $n$  numere naturale.

a) [0.5p] Să se scrie o funcție fără parametri numită **citire\_numere** care să citească datele din fișier și să returneze un obiect de tip list reprezentând matricea de dimensiuni  $n \times n$  care conține numerele în ordinea din fișier. Pentru fișierul din exemplu matricea returnată va fi  $[[2, 10, 6, 4], [6, 7, 9, 3], [4, 7, 4, 10], [10, 4, 6, 7]]$ .

b) [1p] Fie **M** matricea returnată de **citire\_numere**. Se citește de la tastatură un număr **k**. Să se determine mulțimea numerelor (elementelor) din **M** care apar în cel puțin **k** dintre liniile matricei. Să se scrie în fișierul **numere.out** aceste numere ordonate descrescător, câte unul pe linie. Dacă nu există numere cu proprietatea cerută se va scrie în fișier mesajul "Imposibil!".

c) [2p] Să se scrie o funcție **insereaza\_zerouri** care primește 2 parametri:

- o matrice de numere **matrice\_numere** (de tip list)
- un număr natural **x**

Funcția modifică matricea **matrice\_numere** astfel:

- pe fiecare linie din matrice, se **inserează după** fiecare element care este divizibil cu  $x+i$ , unde **i** este indicele liniei (numerotarea liniilor începe de la 0), un nou element cu valoarea 0, apoi
- se șterg din matrice acele linii care conțin un număr egal de 0-uri și de elemente nenule.

d) [0.5p] Să se apeleze funcția **insereaza\_zerouri** pentru matricea returnată de **citire\_numere** și valoarea  $x = 2$  și să se afișeze pe ecran matricea astfel obținută. Numerele de pe o linie a matricei se vor afișa pe câte o linie, separate cu spațiu.

numere.in	Punctul b) k = 2 numere.out	Punctul d) ecran
2 10 6 4 6 7 9 3 4 7 4 10 10 4 6 7	10 7 6 4	6 0 7 9 0 3 0 4 0 7 4 0 10 10 0 4 6 7

## Subiectul 2 – 5 p

DJ Vasile a pregătit o listă de melodii pentru un eveniment la care va participa. Pentru fiecare melodie a notat genul, artistul, titlul, durata (*minute:secunde*) și a grupat cântecele în funcție de genul lor. Fișierul *melodii.in* conține aceste informații sub următoarea formă: fiecare secțiune din fișier începe cu o linie de forma

**Gen >> genul\_melodiei**

apoi urmează mai multe linii, câte una pentru fiecare melodie aparținând aceluși gen, de forma **titlu / artist / minute:secunde**, unde *titlu* și *artist* sunt șiruri de caractere care pot fi formate din mai multe cuvinte separate prin câte un spațiu, dar pot conține și semne de punctuație, iar *minute* și *secunde* sunt formate din câte două cifre.

Un exemplu de astfel de fișier este:

<pre>melodii.in Gen &gt;&gt; Rock Alice is back now / Alice / 03:00 Hello, world! / Alice / 03:00 How are you? / Bob / 02:20 Tomorrow / David Jack / 04:00 Dance with me / Bob / 02:20 Gen &gt;&gt; Hip-hop It wasn't me! / TJ / 02:20 Monkey business / Bruce K / 03:00 Dedication to my enemies / TJ / 03:00 Gen &gt;&gt; Rhythm and blues Brian is coming... / Big Brian / 03:10 Guess who! / Big Brian / 02:40</pre>
--

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 *playlist* care primește următorii parametri (în această ordine):

- structura în care s-au memorat datele la cerința a),
- un număr variabil de parametri reprezentând genuri muzicale,
- un parametru *durata\_minima* cu valoare implicită 02:00
- un parametru *durata\_maxima* cu valoare implicită 03:30

Funcția returnează o listă de tupluri cu elemente de forma (gen, artist, titlu, durata) pentru acele melodii care au durata cuprinsă între *durata\_minima* și *durata\_maxima* și aparțin unuia dintre genurile muzicale primite ca parametri.

Lista returnată va fi ordonată descrescător după *durată*, apoi, în caz de egalitate, crescător după *artist* și apoi după *titlu*.

Să se apeleze funcția *playlist* pentru genurile "Rock" și "Hip-hop" și să se afișeze pe ecran lista returnată.

**Explicații:** pentru datele din fișier lista returnată va conține următoarele elemente:

('Rock', 'Alice is back now', 'Alice', '03:00')  
( 'Rock', 'Hello, world!', 'Alice', '03:00')  
( 'Hip-hop', 'Monkey business', 'Bruce K', '03:00')  
( 'Hip-hop', 'Dedication to my enemies', 'TJ', '03:00')



('Rock', 'Dance with me', 'Bob', '02:20')  
('Rock', 'How are you?', 'Bob', '02:20')  
('Hip-hop', 'It wasn't me!', 'TJ', '02:20')

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

- structura în care s-au memorat datele la cerința a),
- un șir de caractere reprezentând *genul* unei melodii,
- un șir de caractere reprezentând *titlul* unei melodii,
- un șir de caractere reprezentând *artistul*,
- un șir de caractere de forma *minute:secunde* reprezentând *durata* melodiei.

Funcția va adăuga în structură un nou element corespunzător informațiilor date (gen, titlu, artist, durata) și va returna un mesaj de forma "Genul ... contine acum ... melodii.". În cazul în care *genul* primit ca parametru nu exista deja în structură, se va returna mesajul "Nu exista acest gen muzical.", iar structura va rămâne nemodificată.

Se citesc de la tastatură, pe rânduri diferite, informațiile reprezentând *genul*, *titlul*, *artistul* și *durata* unei melodii. Să se apeleze funcția **adauga\_melodie** pentru a adăuga în structură, dacă se poate, informațiile citite de la tastatură și să se afișeze rezultatul funcției (mesajul). După apelul funcției, să se afișeze structura în care s-au memorat datele.

De exemplu, pentru datele de intrare

*Hip-hop*

*Meet me there*

*DGK*

*03:15*

se va actualiza structura și se va afișa mesajul "Genul Hip-hop contine acum 4 melodii.",

iar pentru datele de intrare

*Pop*

*Nice to meet you*

*Aron B*

*02:45*

structura va rămâne nemodificată și se va afișa mesajul "Nu exista acest gen muzical."

## MODALITATEA DE DESFĂȘURARE A TESTULUI DE LABORATOR LA DISCIPLINA "PROGRAMAREA ALGORITMILOR"

- Testul de laborator la disciplina "Programarea algoritmilor" se va desfășura în ziua de **13.01.2024**, în două runde, între orele 9<sup>00</sup> și 11<sup>00</sup>, respectiv 11<sup>30</sup> și 13<sup>30</sup>, astfel:

### ➤ Prima rundă

- 09<sup>00</sup> – 09<sup>15</sup>: efectuarea prezenței studenților
- 09<sup>15</sup> – 10<sup>45</sup>: desfășurarea testului
- 10<sup>45</sup> – 11<sup>00</sup>: trimiterea surselor folosind un formular Google dedicat

### ➤ A doua rundă

- 11<sup>30</sup> – 11<sup>45</sup>: efectuarea prezenței studenților
- 11<sup>45</sup> – 13<sup>15</sup>: desfășurarea testului
- 13<sup>15</sup> – 13<sup>30</sup>: 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](http://old.fmi.unibuc.ro/ro/pdf/2015/consiliu/Regulament_etica_FMI.pdf)).