

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

- Testul de laborator la disciplina "Programarea algoritmilor" se va desfășura în ziua de **08.01.2022**, între orele 9<sup>30</sup> și 12<sup>00</sup>, astfel:
  - **09<sup>30</sup> – 10<sup>00</sup>**: efectuarea prezenței studenților
  - **10<sup>00</sup> – 11<sup>30</sup>**: desfășurarea testului
  - **11<sup>30</sup> – 12<sup>00</sup>**: verificarea faptului că sursele trimise de către studenți au fost salvate pe platformă
- Testul se va desfășura pe platforma MS Teams, iar pe tot parcursul desfășurării lui studenții trebuie să fie conectați pe canalul dedicat **cursului** de "Programarea algoritmilor" corespunzător seriei lor.
- În momentul efectuării prezenței, fiecare student trebuie să aibă pornită camera video în MS Teams și să prezinte buletinul sau cartea de identitate. Dacă dorește să-și protejeze datele personale, studentul poate să acopere codul numeric personal și/sau adresa!
- În timpul desfășurării testului studenții pot să închidă camera video, dar trebuie să o deschidă dacă li se solicită acest lucru de către un cadru didactic!
- Testul va conține **3 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, următoarele informații: numele și prenumele studentului, grupa sa și enunțul subiectului rezolvat în fișierul sursă respectiv. Dacă un student nu reușește să rezolve deloc un anumit subiect, totuși 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)).

### Subiect 1

[4 p.] Fișierul text *text.in* conține, pe mai multe linii, un text în care cuvintele sunt despărțite prin spații și semnele de punctuație uzuale. Să se scrie în fișierul text *text.out* literele din textul dat care au frecvențele relative nenule, precum și frecvențele lor relative, conform modelului din exemplul de mai jos. Frecvența relativă a unei litere într-un text este egală cu raportul dintre frecvența sa în textul respectiv și numărul total de litere din acel text. Literele vor fi scrise în fișierul *text.out* în ordinea descrescătoare a frecvențelor lor, iar în cazul unor frecvențe egale vor fi ordonate alfabetic. Nu se va face distincție între litere mici și litere mari.

#### Exemplu:

text.in	text.out
Ana are multe pere si mere, dar are mai multe mere decat pere si prune mai multe decat pere.	e: 0.250 a: 0.125 r: 0.125 m: 0.097 t: 0.069 i: 0.056 p: 0.056 u: 0.056 d: 0.042 l: 0.042 c: 0.028 n: 0.028 s: 0.028

Textul din fișierul de intrare *text.in* conține 72 de litere mari și mici, iar litera 'e' apare de 18 ori în textul dat, deci frecvența relativă a literei 'e' este egală cu  $18 / 72 = 0.250$ .

## Subiectul 2

Se dă fișierul "**matrice.in**" cu următoarea structură: pe linia  $i$  se află separate prin câte un spațiu  $n$  numere naturale reprezentând elementele de pe linia  $i$  a unei matrice, ca în exemplul de mai jos. Este cunoscut faptul că în fișier se află  $n * n$  elemente numere naturale, unde  $n$  este un număr natural impar  $> 2$ .

**a) [0,25p]** Să se scrie o funcție **citire\_matrice** care citește datele din fișierul "**matrice.in**" și returnează o matrice de dimensiune  $n \times n$  formată din aceste numere.

**b) [1,5p]** Să se scrie o funcție care primește ca parametru matricea și returnează matricea bordată după următoarele reguli:

- se va adăuga o coloană nouă la final (pe poziția  $n$ ) care va avea pe poziția  $k$  **suma** valorilor de pe **linia**  $k$
- se va adăuga o linie nouă la final (poziția  $n$ ) care va avea pe poziția  $k$  **suma** valorilor de pe **coloana**  $k$

**c) [1,25p]** Se consideră matricea citită la punctul **a)**, peste care se aplică funcția de la punctul **b)**. Să se parcurgă matricea mai întâi pe diagonala principală, apoi pe diagonala secundară și, în final, restul elementelor care **nu** aparțin diagonalelor (parcurea se face pe linii de sus în jos și de la stânga la dreapta) și se afișează elementele în fișierul "**matrice.out**".

matrice.in	după bordare	matrice.out
7 8 9 6 1 2 5 4 3	7 8 9 24 6 1 2 9 5 4 3 12 18 13 14 45	7 1 3 45 24 2 4 18 8 9 6 9 5 12 13 14

### Subiect 3

Fișierul text *drumuri.in* conține informații despre drumurile dintre orașele unei țări. O linie din fișier are următoarea structură:

***Nume\_Oras\_1 - Nume\_Oras\_2 distanta stare\_drum***

unde ***Nume\_Oras\_1*** și ***Nume\_Oras\_2*** sunt numele a două orașe (un nume este un șir de cuvinte separate prin câte un spațiu), ***distanta*** este lungimea drumului dintre cele două orașe, iar ***stare\_drum*** (număr natural) este un număr natural între 0 și 5 reprezentând calitatea drumului între cele două orașe. Pe un drum se poate circula doar într-un sens, respectiv de la ***Nume\_Oras\_1*** la ***Nume\_Oras\_2***.

Un exemplu de astfel de fișier este:

drumuri.in
Oraselul Mic - Satul Mare 20 5
Oraselul Mic - Moeni 10 1
Satul Mare - Capitala 100 2
Satul Mare - Pol 20 5
Capitala - Pol 23 3

- a) [2 p.] Să se memoreze datele din fișier într-o singură structură de date astfel încât să se **răspundă eficient** la cerințele de la punctele următoare (interogarea și modificarea informațiilor despre un drum, determinarea orașelor accesibile din alt oraș).
- b) [1 p.] Scrieți o funcție **modifica\_stare** care are următorii parametri (în această ordine):
- structura în care s-au memorat datele la cerința a)
  - un număr natural *s* între 0 și 5 reprezentând starea unui drum
  - două șiruri de caractere *o1* și *o2*; ultimul parametru *o2* are valoarea implicită șirul vid. Dacă *o2* este un șir nevid, funcția va modifica starea drumului de la orașul cu numele *o1* la orașul *o2* cu valoarea *s*, dacă acest drum există. Dacă *o2* este șirul vid funcția va modifica starea tuturor drumurilor de la orașul *o1* la celelalte orașe în *s*. Funcția va returna numărul de drumuri a căror stare a fost modificată.
- c) [1 p.] Scrieți o funcție **accesibil** cu 2 parametri: structura în care s-au memorat datele la cerința a) și un număr variabil de șiruri de caractere reprezentând nume de orașe și returnează mulțimea orașelor la care se poate ajunge din cel puțin unul dintre orașele primite ca parametru folosind **unul** dintre drumurile din oraș. Apelați funcția pentru orașele *Oraselul Mic* și *Capitala* și afișați rezultatul obținut (ordinea orașelor din mulțimea returnată nu contează).

ieșire
{ 'Satul Mare', 'Moeni', 'Pol' }

**Explicații:** Din *Oraselul Mic* putem ajunge în *Satul Mare* și *Moeni* (folosind un drum din rețea), iar din *Capitala* în *Satul Mare* și *Pol*.