

## 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 *numere.in* conține, pe fiecare linie, câte un șir de numere întregi despărțite prin spații. Să se scrie în fișierul text *numere.out* șirurile din fișierul de intrare grupate în funcție de suma elementelor lor, conform modelului din exemplul de mai jos. Grupele de șiruri vor fi scrise în ordinea crescătoare a sumelor elementelor lor, iar în fiecare grupă șirurile se vor scrie în ordinea descrescătoare a numărului de elemente.

**Exemplu:**

numere.in	numere.out
10 -5 5 10 100 -20 -30 11 9 5 5 10 20 10 -100 -100	Suma -200: -100 -100 Suma 20: 10 -5 5 10 11 9 Suma 50: 5 5 10 20 10 100 -20 -30

## Subiectul 2

a) [0,5p] Scrieți o funcție **citire\_matrice** care primește un parametru reprezentând numele unui fișier care conține elementele unei matrice  $m \times n$  de numere întregi cu următoarea structură:

- pe prima linie a fișierului sunt  $m$  și  $n$
- pe a doua linie sunt  $m * n$  numere separate prin câte un spațiu reprezentând elementele matricei transformate în vector prin concatenarea liniilor matricei de la prima la ultima; astfel primele  $n$  numere sunt elementele primei linii din matrice, urmate de elementele celei de a doua linii etc.

Funcția citește elementele matricei din fișierul cu numele dat ca parametru și returnează matricea cu aceste elemente. Dacă în fișierul de intrare nu sunt  $m \times n$  elemente pe linia a doua funcția va returna o listă vidă. Pentru fișierul "**matricemn.in**" din exemplul de mai jos matricea este:

```
1 2
3 4
5 6
```

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

- o matrice (listă de liste)
- un număr variabil de numere naturale distincte reprezentând indici ai liniilor din matrice (indicele primei linii din matrice este 0)
- un parametru *tip* care poate primi valoarea 0 sau 1, implicit având valoarea 0.

Funcția adaugă la finalul matricei, dacă *tip* = 0, sau la începutul matricei, dacă *tip* = 1, o linie egală cu suma liniilor cu indicii dați ca parametru (suma a două linii se face element cu element). Funcția va modifica matricea primită ca parametru.

c) [1p] Se dă fișierul "**matricemn.in**" cu structura descrisă la punctul a) ( $m > 2$ ). Folosind apeluri utile ale funcțiilor de la a) și b) să se citească matricea din fișierul "**matricemn.in**" și să se modifice această matrice inserând la începutul matricei o linie cu suma dintre prima și a doua linie din matrice și la sfârșitul matricei o linie cu suma dintre penultima și ultima linie din matrice. Să se afișeze pe ecran matricea obținută astfel încât elementele de pe fiecare coloană să fie aliniate la dreapta, știind că numerele din matrice pot avea maxim 4 cifre.

matricemn.in	Iesire pe ecran	
3 2	4	6
1 2 3 4 5 6	1	2
	3	4
	5	6
	8	10

### 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*.