

## 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 pătratice  $n \times n$  de numere întregi cu următoarea structură:

- pe prima linie a fișierului este  $n$

- pe a doua linie sunt  $n * n$  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 de pe linie 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. Pentru fișierul "**matrice.in**" din exemplul de mai jos matricea este :

1 2 3

4 5 6

7 8 9

**b) [1,5p]** Scrieți o funcție **duplicare** care primește ca parametri (în această ordine): o matrice (listă de liste) și un număr variabil de numere naturale reprezentând indici ai liniilor din matrice (indicele primei linii din matrice este 0) și **inserează după fiecare linie cu indicele dat ca parametru o copie a ei** (duplică linia). Funcția va modifica matricea primită ca parametru.

**c) [1p]** 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**", să se modifice această matrice duplicând prima și a doua linie (după prima linie se va insera o linie egală cu ea, la fel și după a doua) și adăugând apoi 1 la primul element de pe prima linie. Să se afișeze pe ecran matricea obținută.

matrice.in	Iesire pe ecran
3	2 2 3
1 2 3 4 5 6 7 8 9	1 2 3
	4 5 6
	4 5 6
	7 8 9

### Subiect 3

Se consideră o rețea în plan formată din puncte unite prin arce. Fiecare punct are coordonatele întregi, iar un arc are asociată o grosime (număr natural) și o culoare (un șir de caractere fără spații reprezentând numele culorii, de exemplu: roșu, verde, albastru). Un punct cu coordonatele  $x$  și  $y$  va fi notat  $(x,y)$ . Se consideră fișierul text *arce.in* care conține informații despre o astfel de rețea, fiecare linie conținând informații despre un arc sub forma:

**$(x1,y1)-(x2,y2)$  grosime culoare**

unde  $(x1,y1)$  este punctul din care începe arcul,  $(x2,y2)$  este punctul în care se termină arcul, **grosime** reprezintă grosimea arcului, iar **culoare** colorarea sa. Acest arc se va numi arc între punctele  $(x1,y1)$  și  $(x2,y2)$ , vom numi  $(x1,y1)$  capătul inițial al arcului și  $(x2,y2)$  capătul final al arcului. Un exemplu de fișier de acest tip este următorul:

arce.in
(1,2)-(1,3) 10 roșu
(1,4)-(1,2) 5 albastru
(1,3)-(2,6) 5 roșu
(2,6)-(2,7) 10 albastru
(2,7)-(3,8) 4 roșu
(2,1)-(3,8) 4 verde
(3,8)-(1,2) 11 roșu

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

b) [1 p.] Scrieți o funcție **modifica\_arc** care primește 5 parametri:

- în primul parametru se transmite structura în care s-au memorat datele la cerința a)
- următorii 2 parametri  $p1$  și  $p2$  sunt două tupluri cu două elemente reprezentând capătul inițial și final al unui arc
- ultimii 2 parametri sunt un număr natural reprezentând grosimea arcului și un șir de caractere reprezentând culoarea arcului.

Dacă arcul există deja în rețea funcția va modifica informațiile asociate arcului (grosimea și culoarea), altfel funcția va adăuga acest arc la rețea (modificând structura trimisă ca parametru). Funcția va returna **numărul de arce care au capătul inițial în  $p1$** . Să se apeleze funcția pentru a adăuga la rețea arcul de la (5,6) la (7,8) de grosime 5 și culoare verde.

c) [1 p.] Scrieți o funcție **sterge\_punct** care primește 2 parametri: în primul parametru se transmite structura în care s-au memorat datele la cerința a), iar al doilea parametru este un tuplu cu 2 elemente reprezentând un punct din rețea. Funcția va șterge din rețea punctul primit ca parametru (se vor șterge toate arcele care au acel punct ca și capăt inițial sau capăt final) și **va returna mulțimea arcelor din rețea**. Să se apeleze funcția pentru a șterge punctul (1,2) și să se afișeze rezultatul returnat. După ștergere se vor afișa datele memorate în structură în același format în care s-au dat și datele în fișier (nu contează ordinea în care se vor afișa arcele).

ieșire
Arcele ramase dupa stergere:
(1, 3)-(2, 6) 5 roșu
(2, 6)-(2, 7) 10 albastru
(2, 7)-(3, 8) 4 roșu
(2, 1)-(3, 8) 4 verde
(5, 6)-(7, 8) 5 verde