

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 9³⁰ și 12⁰⁰, astfel:
 - **09³⁰ – 10⁰⁰**: efectuarea prezenței studenților
 - **10⁰⁰ – 11³⁰**: desfășurarea testului
 - **11³⁰ – 12⁰⁰**: 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).

Subiect 1

[4 p.] Fișierul text *numere.in* conține, pe mai multe linii, numere naturale despărțite prin spații. Definim *indicele binar* al unui număr natural n ca fiind perechea (x, y) , unde x reprezintă numărul biților nenuli din reprezentarea binară a lui n , iar y reprezintă numărul biților nuli din reprezentarea sa binară. De exemplu, numărul 133 are reprezentarea binară 10000101, deci indicele său binar este (3, 5). Să se scrie în fișierul text *numere.out* numerele din fișierul *numere.in* grupate în funcție de indicii lor binari, conform modelului din exemplul de mai jos. Grupele de numere vor fi scrise în ordinea descrescătoare a numărului de numere pe care le conțin, iar în caz de egalitate se vor scrie în ordinea lexicografică a indicilor lor binari. În cadrul fiecărei grupe numerele vor fi ordonate crescător. Fiecare număr va fi scris o singură dată în fișierul de ieșire.

Exemplu:

numere.in	numere.out
101 133 102	(3, 5): 133,137,152,224
224 152 10	(4, 3): 101,102,108,120
133 133 101 888	(2, 2): 10,12
7 12 543 120 137	(6, 4): 543,888
10 108 108 102	(3, 0): 7

Subiectul 2

Fișierul "**date.in**" are n linii cu următoarea structură: pe linia i sunt prezente, separate prin câte un spațiu, n numere naturale reprezentând elementele de pe linia i dintr-o matrice, ca în exemplu.

Linile și coloanele unei matrice se presupun numerotate de la 0.

a) [0,25p] Scrieți o funcție **citire_matrice** care citește numerele din fișierul "**date.in**" și returnează o matrice de dimensiuni n x n formată din aceste numere.

b) [1,5p] Scrieți o funcție care primește ca parametri: o matrice, un caracter ch și doi parametri x și y cu valoare implicită 0.

- Dacă la apel al doilea parametru - caracterul ch primește valoarea "**d**", funcția nu va primi decât 2 parametri și trebuie să interschimbe elementele de pe diagonala principală cu elementele de pe diagonala secundară.
- Dacă la apel al doilea parametru - caracterul ch - primește valoarea "**l**", funcția va primi valori și pentru **parametrii** x, y și trebuie să interschimbe linia x cu linia y.

c) [1,25p] Folosind apeluri ale funcției definite la punctul **b)**, **oglindiți** matricea returnată de funcția de la punctul a) după linia de pe poziția $[n / 2]$ și interschimbați elementele de pe diagonala principală cu cele de pe diagonala secundară. După oglindire și interschimbare, să se parcurgă matricea în **zig-zag** pe coloane și să se afișeze șirul obținut în fișierul "**date.out**", ca în exemplu. Se cunoaște faptul că **n este impar**.

Explicație suplimentară : Parcurgerea în zig-zag pe coloane se va face de la stanga la dreapta, astfel:

- prima coloană se parcurge de jos în sus,
- a doua coloană se parcurge de sus în jos,
- a treia coloană se parcurge de jos în sus etc.

date.in	după oglindire și inteschimbare	date.out
21 10 11 20 1	5 6 15 16 25	1 2 3 4 5 6 7 8 9 10 11 12 13 14
2 19 12 9 22	4 7 14 17 24	15 16 17 18 19 20 21 22 23 24 25
3 8 13 18 23	3 8 13 18 23	
4 17 14 7 24	2 9 12 19 22	
25 6 15 16 5	1 10 11 20 21	

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*.