

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 *teatru.in* conține, pe mai multe linii, un fragment dintr-o piesă de teatru, respectiv pe fiecare linie se află câte o replică a unui personaj, sub forma *personaj: replică*. Numele unui personaj poate fi format din mai multe cuvinte, iar o replică nu va conține niciodată caracterul ':'. Să se scrie în fișierul text *teatru.out* cuvintele din fragmentul dat grupate în funcție de numărul personajelor care le-au pronunțat, conform modelului din exemplul de mai jos. Cuvintele vor fi scrise în ordinea descrescătoare a numărului de personaje care le-au pronunțat, iar în caz de egalitate se vor scrie în ordine alfabetică. Pentru fiecare cuvânt, numele personajele care l-au pronunțat vor fi ordonate alfabetic. Fiecare cuvânt va fi scris o singură dată și nu se va face distincție între litere mici și litere mari.

Exemplu:

teatru.in	teatru.out
Tipatescu: Misel! Pristanda: Curat misel! Tipatescu: Murdar! Pristanda: Curat murdar! Tipatescu: Ei! Nu s-alege! Pristanda: Nu s-alege! Farfuridi: Vrei sa vorbesc curat si deslusit, stimabile?	curat: Farfuridi,Pristanda misel: Pristanda,Tipatescu murdar: Pristanda,Tipatescu nu: Pristanda,Tipatescu s-alege: Pristanda,Tipatescu deslusit: Farfuridi ei: Tipatescu sa: Farfuridi si: Farfuridi stimabile: Farfuridi vorbesc: Farfuridi vrei: Farfuridi

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

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