

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 *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

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

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