

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. Pentru fiecare cuvânt distinct din textul dat să se scrie în fișierul text *text.out* pozițiile pe care acesta apare în text. Poziția unui cuvânt *w* va fi scrisă în fișier sub forma unei perechi (*x,y*) în care *x* reprezintă numărul de ordine al liniei pe care apare cuvântul, iar *y* reprezintă numărul de ordine al lui *w* pe linia *x*. Liniile din fișierul *text.in* se consideră numerotate de la 1 de sus în jos, iar pe o linie cuvintele sunt numerotate începând cu 1 de la stânga spre dreapta. Cuvintele vor fi scrise în fișierul *text.out* în ordinea descrescătoare a frecvențelor lor în întreg textul, 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.	multe: (1,3) (2,4) (3,6) pere: (1,4) (3,2) (3,8) are: (1,2) (2,2) decat: (3,1) (3,7) mai: (2,3) (3,5) mere: (1,6) (2,5) si: (1,5) (3,3) ana: (1,1) dar: (2,1) prune: (3,4)

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 indicii 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 linii numite legături (muchii). Fiecare punct are coordonatele întregi, iar o legătură are asociată o culoare (un șir de caractere fără spații reprezentând numele culorii, de exemplu: roșu, verde, albastru) și o etichetă (un șir care poate conține spații). Un punct cu coordonatele x și y va fi notat (x,y) . Se consideră fișierul text *legaturi.in* care conține informații despre o astfel de rețea, fiecare linie conținând informații despre o legătură sub forma:

[x1,y1] [x2,y2] culoare eticheta

unde $[x1,y1]$ și $[x2,y2]$ sunt punctele între care există legătură, iar **culoare** și **etichetă** reprezintă culoarea și eticheta asociate legăturii. Vom spune că legătura este între punctele $[x1,y1]$ și $[x2,y2]$, vom numi punctele $[x1,y1]$ și $[x2,y2]$ capetele legăturii (legătura este neorientată și bidirecțională) și cele două puncte sunt vecine. Un exemplu de fișier de acest tip este următorul:

legaturi.in		
[1,2]	[1,3]	rosu legatura 1
[1,2]	[1,4]	albastru legatura 2
[1,3]	[2,6]	rosu legatura 3
[2,6]	[2,7]	albastru legatura 4
[2,7]	[3,8]	rosu legatura 5

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 (interogarea și modificarea informațiilor despre o legătură dintre două puncte date, determinarea vecinilor unui punct).

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

- în primul parametru se transmite structura în care s-au memorat datele la cerința a)
- următorii 2 parametri sunt tupluri cu 2 elemente reprezentând capetele unei legături
- ultimii 2 parametri sunt șiruri de caractere reprezentând culoarea și eticheta unei legături.

Dacă există deja o legătură în rețea între punctele primite ca parametru funcția va returna **False**, altfel funcția va adăuga această legătură în rețea (modificând structura trimisă ca parametru) și va returna **True**. Să se apeleze funcția pentru punctele (1,3) și (2,7), culoarea negru și eticheta "legatura noua" și să se afișeze legăturile memorate în structura obținută în același format în care s-au dat și datele în fișier (nu contează ordinea în care se vor afișa legăturile).

ieșire		
[1,2]	[1,3]	rosu legatura 1
[1,2]	[1,4]	albastru legatura 2
[1,3]	[2,6]	rosu legatura 3
[2,6]	[2,7]	albastru legatura 4
[2,7]	[3,8]	rosu legatura 5
[1,3]	[2,7]	negru legatura noua

c) [1 p.] Scrieți o funcție **vecini** care primește ca parametri (în această ordine): structura în care s-au memorat datele la cerința a) și un număr variabil de tupluri cu 2 elemente reprezentând puncte din rețea. Funcția va returna o listă cu acele puncte p din rețea cu proprietatea că există legătură de la p la orice punct primit ca parametru (vecinii comuni pentru punctele primite ca parametru). Punctele din lista returnată vor fi ordonate descrescător după a doua coordonată. Apelați funcția pentru punctele (2,7) și (1,2) și afișați rezultatul obținut.

ieșire
[(3, 8), (1, 3)]