

**MODALITATEA DE DESFĂȘURARE A EXAMENULUI  
LA DISCIPLINA "PROGRAMAREA ALGORITMILOR"  
DIN SESIUNEA DE RESTANȚE 31.05.2022 – 06.06.2022**

- Examenul la disciplina "Programarea algoritmilor" se va desfășura în ziua de 03.06.2022, între orele 9<sup>00</sup> și 14<sup>00</sup>, astfel:
  - 09<sup>00</sup> – 9<sup>15</sup>: efectuarea prezenței studenților la testul de laborator
  - 09<sup>15</sup> – 10<sup>45</sup>: desfășurarea testului de laborator
  - 10<sup>45</sup> – 11<sup>00</sup>: verificarea faptului că sursele trimise de către studenți au fost salvate pe platformă
  - 11<sup>00</sup> – 11<sup>30</sup>: pauza
  - 11<sup>30</sup> – 11<sup>45</sup>: efectuarea prezenței studenților la examenul scris
  - 11<sup>45</sup> – 13<sup>45</sup>: desfășurarea examenului scris
  - 13<sup>45</sup> – 14<sup>00</sup>: verificarea faptului că fișierele trimise de către studenți au fost salvate pe platforma
- Ambele probe se vor desfășura pe platforma MS Teams, iar pe tot parcursul desfășurării lor studenții trebuie să fie conectați pe canalul dedicat cursului de "Programarea algoritmilor" corespunzător seriei lor (pe întâlnirea special creată pentru examen).
- 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!

**Precizări privind desfășurarea testului de laborator:**

- 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, numele și prenumele studentului, precum și grupa sa. 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)).
- Operațiile de sortare se vor efectua folosind funcții sau metode predefinite din limbajul Python.

### Precizări privind desfășurarea examenului scris:

- Toate subiectele se vor rezolva folosind limbajul Python.
  - Subiectul 1 este obligatoriu, iar dintre subiectele 2, 3 și 4 se vor rezolva CEL MULT DOUĂ, la alegere.
  - Citirea datelor de intrare se va realiza de la tastatură, iar rezultatele vor fi afișate pe ecran.
  - Se garantează faptul că datele de intrare sunt corecte.
  - Operațiile de sortare se vor efectua folosind funcții sau metode predefinite din limbajul Python.
  - Rezolvările subiectelor alese dintre subiectele 2, 3 și 4 trebuie să conțină:
    - o scurtă descriere a algoritmului și o argumentare a faptului că acesta se încadrează într-o anumită tehnică de programare;
    - în cazul problemelor rezolvate folosind metoda Greedy sau metoda programării dinamice se va argumenta corectitudinea criteriului de selecție sau a relațiilor de calcul;
    - în cazul subiectelor unde se precizează complexitatea maximă pe care trebuie să o aibă soluția, se va argumenta complexitatea soluției propuse și vor primi punctaj maxim doar soluțiile corecte care se încadrează în complexitatea cerută;
    - în fiecare program Python se va preciza, pe scurt, sub forma unor comentarii, semnificația variabilelor utilizate.
  - Pentru subiectele 1 nu contează complexitățile soluțiilor propuse.
  - Rezolvările corecte care nu respectă restricțiile indicate vor primi punctaje parțiale.
  - Se acordă 1 punct din oficiu.
- 
- Rezolvările tuturor subiectelor se vor scrie de mână, folosind pix/stilou cu culoarea pasteii/cernelii albastră sau neagră. Pe fiecare pagina studentul își va scrie numele și grupa, iar paginile trebuie să fie numerotate.
  - Înainte de expirarea timpului alocat examenului, toate paginile vor fi fotografiate/scanate clar, în ordinea corectă, și transformate într-un singur fișier PDF care va fi încărcat pe platforma MS Teams folosind un anumit formular.
- 
- Numele fișierului PDF **trebuie să respecte șablonul** *grupa\_nume\_prenume.pdf*. 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.pdf*.

### Subiectul 1 (4 puncte)

**[4 p]** Fișierul text *numere.in* conține, pe mai multe linii, numere naturale despărțite prin spații. Să se scrie în fișierul text *numere.out* numerele din fișierul *numere.in* care au fost modificate prin inserarea simbolului delimitator “-” astfel încât în fiecare bucată nou obținută cifrele să fie în ordine descrescătoare. De exemplu, numărul 5110472633 va fi modificat în 5110-4-72-633. Numerele astfel modificate vor fi scrise în fișier conform modelului din exemplul de mai jos, grupate crescător în funcție de numărul de părți în care a fost împărțit numărul. În fiecare grupă, se menționează numărul de părți apoi se enumeră numerele modificate corespunzătoare, în ordine descrescătoare a valorilor numerelor (pentru a stabili ordinea valorilor se compară numerele de dinainte de modificare). Fiecare număr va fi scris o singură dată.

#### Exemplu:

<code>numere.in</code>	<code>numere.out</code>
5110472633 34251 987 52311 221683 987 47882 84110 65592	1 parti: 84110, 987 2 parti: 655-92, 52-311 3 parti: 221-6-83, 4-7-882, 3-42-51 4 parti: 5110-4-72-633

## Subiectul 2 (3 puncte)

**a) [0,5 p]** Fișierul "matrice.in" are  $n > 1$  linii care conțin fiecare câte  $n$  numere naturale nenule separate prin câte un spațiu. Toate numerele din fișier sunt distincte.

Să se scrie o funcție fără parametri numită **citire\_date** care să citească datele din fișier și să returneze matricea de dimensiuni  $n \times n$  care conține numerele în ordinea din fișier.

**b) [1,5 p]** Să se scrie o funcție **diferente** care primește ca parametri o matrice pătratică  $n \times n$  și un număr variabil de parametri cu valori cuprinse între 0 și  $n-1$ , reprezentând indicii unor linii. Funcția va modifica matricea primită ca parametru astfel: pe fiecare linie al cărei indice a fost primit ca parametru, se va adăuga la final un nou element a cărui valoare este egală cu diferența dintre cel mai mare și cel mai mic număr de pe acea linie.

**c) [1 p]** Să se apeleze funcția de la **b)** pentru matricea obținută la **a)** și indicii primelor două linii, apoi să se tripleze valoarea primului element de pe ultima linie. Matricea astfel modificată să se afișeze pe ecran, fără paranteze și virgule, elementele de pe fiecare linie să fie separate prin câte un spațiu.

### Exemplu:

matrice.in	Afișare pe ecran
11 20 53 16 22 70 35 67 97 43 54 14 2 8 19 42 64 88 31 50 33 95 17 29 40	11 20 53 16 22 42 70 35 67 97 43 62 54 14 2 8 19 42 64 88 31 50 99 95 17 29 40

### Subiectul 3 (4 puncte)

Fișierul "carti.in" conține rezultatele unui sondaj făcut asupra unor cărți în rândul cititorilor. Fiecare cititor a fost identificat printr-un cod și a acordat note de la 1 la 5 cărților pe care a dorit să le evalueze. O linie din fișier are următoarea structură:

**nume\_carte cod\_cititor nota**

unde **nume\_carte** este numele unei cărți (format din cuvinte separate prin câte un spațiu)

**cod\_cititor** este un șir de caractere fără spații reprezentând codul unui cititor, iar **nota** este un număr natural între 1 și 5 reprezentând nota dată de utilizator cărții.

Un exemplu de astfel de fișier este:

carti.in
Algorithm Design c1 4
Introduction to Algorithms c3 5
Algorithm Design c2 5
Lista exmatriculati c5 1
Introduction to Algorithms c2 4
Lista exmatriculati c2 1
Lista exmatriculati c3 1
Introduction to Algorithms c5 3
Graph algorithms with Applications c1 3
Graph algorithms with Applications c2 4
Lista exmatriculati c1 1
Lista exmatriculati c4 1

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

b) [1 p.] Scrieți o funcție **sterge\_nota** care are următorii parametri (în această ordine):

- structura în care s-au memorat datele la cerința a)
- un șir de caractere *nume\_carte* reprezentând numele unei cărți
- un șir de caractere *cod* reprezentând codul unui cititor

Funcția șterge din structura de date nota dată de cititorul cu codul *cod* cărții *nume\_carte* (dacă cititorul a dat notă cărții *nume\_carte*). **Funcția va returna media aritmetică a notelor primite de carte după această actualizare** (0 dacă nu are note).

Să se apeleze această funcție pentru un nume de carte și un cod de cititor citite de la tastatură și să se afișeze rezultatul returnat de funcție și structura în care s-au memorat datele după apelul funcției.

c) [1 p.] Spunem că un cititor **apreciază** o carte dacă îi dă o notă mai mare sau egală cu 3. Scrieți o funcție **aprecieri** care primește următorii parametri: structura în care s-au memorat datele la cerința a) și un număr variabil de șiruri de caractere reprezentând nume de cărți și returnează o listă de tupluri de forma (*cod\_cititor*, *numar\_carti*, *nota\_maxima*) pentru toți cititorii care au apreciat **cel puțin una** dintre cărțile primite ca parametru; semnificația componentelor tuplului este: *cod\_cititor* este codul unui cititor care a apreciat cel puțin o carte dintre cele primite ca parametru, *numar\_carti* reprezintă câte dintre aceste cărți a apreciat cititorul, iar *nota\_maxima* este cea mai mare notă pe care a dat-o unei astfel de cărți. Lista returnată va fi ordonată descrescător după numărul de cărți apreciate, în caz de egalitate, descrescător după nota maximă acordată, iar apoi, în caz de egalitate, după codul cititorului. Să se apeleze funcția pentru cărțile *Algorithm Design*, *Introduction to Algorithms*, *Lista exmatriculati* și să se afișeze lista returnată. **Explicații:** pentru datele din fișier lista returnată va fi [(*'c2'*, 2, 5), (*'c3'*, 1, 5), (*'c1'*, 1, 4), (*'c5'*, 1, 3)] deoarece, dintre cărțile trimise ca parametru, cititorul c2 apreciază cărțile *Algorithm Design*, *Introduction to Algorithms* (nu și cartea *Lista exmatriculati*) și cea mai mare notă dată a fost 5 pentru *Algorithm Design*, cititorul c3 a apreciat (a dat nota 5, mai mare sau egală cu 3) doar cartea *Introduction to Algorithms* din lista de parametri etc.