

**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 pastei/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 *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* cuvintele din fișierul *text.in* care au fost modificate prin inserarea simbolului delimitator “.” astfel încât să separe vocalele de consoanele aflate pe poziții alăturate în cuvânt. De exemplu, cuvântul “**f**ru**c**te” va fi modificat în “**f**r.**u**.**c**t.**e**”. Cuvintele astfel modificate vor fi scrise în fișier conform modelului din exemplul de mai jos, grupate descrescător în funcție de numărul de părți în care a fost împărțit cuvântul. În fiecare grupă, se menționează numărul de părți apoi se enumeră cuvintele modificate corespunzătoare, în ordine alfabetică (pentru a stabili ordinea alfabetică se compară cuvintele de dinainte de modificare). Fiecare cuvânt va fi scris o singură dată și nu se va face distincție între litere mici și litere mari.

### Exemplu:

text.in	text.out
Oana are multe fructe coapte de cules. Impreuna cu surioara ei va pregati dulceata de caise cu prune, apoi placinte cu gutui.	6 parti: d.u.lc.ea.t.a, pl.a.c.i.nt.e, pr.e.g.a.t.i, s.u.r.ioa.r.a 5 parti: c.u.l.e.s, i.mpr.eu.n.a 4 parti: c.ai.s.e, c.oa.pt.e, fr.u.ct.e, g.u.t.ui, m.u.lt.e, pr.u.n.e 3 parti: a.p.oi, a.r.e, oa.n.a 2 parti: c.u, d.e, v.a 1 parti: ei

## 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\_matrice** 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 **mutare** care primește ca parametri o matrice pătratică de dimensiuni  $n \times n$  și un număr variabil de parametri cu valori cuprinse între  $0$  și  $n-1$ , reprezentând indicii unor coloane. Funcția va modifica matricea primită ca parametru astfel: pe fiecare coloană al cărei indice a fost primit ca parametru, se vor interschimba pozițiile în care se află cea mai mică și cea mai mare valoare de pe acea coloană.

**c) [1 p]** Să se apeleze funcția de la **b)** pentru matricea obținută la **a)** și indicii primelor două coloane, apoi să se dubleze valoarea ultimului element de pe prima 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	<b>70</b> 20 53 16 <b>44</b>
70 35 67 97 43	<b>11</b> 35 67 97 43
54 14 2 8 19	54 <b>95</b> 2 8 19
42 64 88 31 50	42 64 88 31 50
33 95 17 29 40	33 <b>14</b> 17 29 40

### Subiectul 3 (4 puncte)

Fișierul "**filme.in**" conține rezultatele unui sondaj făcut asupra unor filme în rândul utilizatorilor. Fiecare utilizator a acordat note de la 1 la 5 filmelor pe care a dorit să le evalueze. O linie din fișier are următoarea structură:

**id\_utilizator nume\_film nota**

unde **id\_utilizator** este un șir de caractere fără spații reprezentând numele unui utilizator, **nume\_film** este numele unui film (format din cuvinte separate prin câte un spațiu) iar **nota** este un număr natural între 1 și 5 reprezentând nota dată de utilizator filmului.

Un exemplu de astfel de fișier este:

filme.in		
ut1	Doua lozuri	5
ut2	Morometii	4
ut2	Doua lozuri	5
ut1	Extemporal la dirigentie	4
ut1	Ciresarii	4
ut1	Extemporal la dirigentie	3
ut1	La facultate	3
ut3	La facultate	3
ut1	La restanta	1
ut3	Doua lozuri	5
ut2	Ciresarii	5
ut2	La restanta	3

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 **modifica\_nota** care are următorii parametri (în această ordine):

- structura în care s-au memorat datele la cerința a)
- un șir de caractere *film* reprezentând numele unui film
- un șir de caractere *id* reprezentând un utilizator
- un număr natural *nota\_noua* între 1 și 5 reprezentând o notă

Dacă utilizatorul *id* a dat deja o notă filmul *film*, atunci se va actualiza această notă cu valoarea parametrului *nota\_noua*. Altfel, se va adăuga în structură pentru filmul *film* nota *nota\_noua* de la utilizatorul *id*.

Să se apeleze această funcție pentru un utilizator, un film și o notă citite de la tastatură și după apelul funcției să se afișeze structura în care s-au memorat datele.

c) [1 p.] Scrieți o funcție **preferinte\_comune** care primește următorii parametri: structura în care s-au memorat datele la cerința a), un număr variabil de șiruri de caractere reprezentând nume de utilizatori și un parametru *nota\_minima* și returnează o listă de tupluri de tip (*nume\_film, medie*) cu filmele care au primit o nota mai mare decât *nota\_minima* **de la toți utilizatorii primiți ca parametru**, unde *nume\_film* este numele unui astfel de film, iar *medie\_film* este un număr real reprezentând media aritmetică a notelor date de utilizatorii primiți ca parametru acestui film. Lista returnată va fi ordonată descrescător după medie și, în caz de egalitate, crescător după nume. Să se apeleze funcția pentru utilizatorii *ut1* și *ut2* și nota minimă **2** și să se afișeze lista returnată. **Explicații:** pentru datele din fișier lista returnată va fi [(*'Doua lozuri'*, 5.0), (*'Ciresarii'*, 4.5)], deoarece filmul *'Doua lozuri'* a primit nota 5 și de la *ut1* și de la *ut2*, iar filmul *'Ciresarii'* a primit notele 4, respectiv 5 de la cei doi utilizatori *ut1* și *ut2*; filmul *'La restanta'* nu a fost inclus în rezultat deoarece nu a primit notă mai mare decât 2 de la ambii utilizatori.