

## 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 9<sup>30</sup> și 12<sup>00</sup>, astfel:
  - **09<sup>30</sup> – 10<sup>00</sup>**: efectuarea prezenței studenților
  - **10<sup>00</sup> – 11<sup>30</sup>**: desfășurarea testului
  - **11<sup>30</sup> – 12<sup>00</sup>**: 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](http://old.fmi.unibuc.ro/ro/pdf/2015/consiliu/Regulament_etica_FMI.pdf)).

### Subiect 1

[4 p.] Fișierul text *text.in* conține pe prima linie un cuvânt *w* nevid format din litere mici ale alfabetului englez, iar pe următoarele linii un text în care cuvintele sunt despărțite prin spații și semnele de punctuație uzuale. Să se scrie în fișierul text *text.out* toate cuvintele din fișierul *text.in* care au mulțimea literelor inclusă în mulțimea literelor cuvântului *w* sau mesajul "Imposibil" dacă în fișierul de intrare nu există nici un cuvânt cu proprietatea cerută, conform modelului din exemplul de mai jos. Cuvintele vor fi scrise grupat, în funcție de mulțimile literelor. Grupele vor fi scrise în ordine lexicografică, iar în cadrul fiecărui grup cuvintele vor fi scrise în ordinea descrescătoare a lungimilor lor. 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
semnificare Mama s-a gandit sa isi puna seara pe masa de ceara o sama de bucate si bauturi care sunt si acre si dulci.	Literele ['a', 'c', 'e', 'r']: ceara care acre Literele ['a', 'e', 'r', 's']: seara Literele ['a', 'm']: mama Literele ['a', 'm', 's']: masa sama Literele ['a', 's']: s-a sa Literele ['i', 's']: isi si

Cuvântul "semnificare" are mulțimea literelor egală cu {'a', 'c', 'e', 'f', 'i', 'm', 'n', 'r', 's'}, iar cuvintele scrise în fișierul *text.out* sunt singurele din fișierul *text.in* care au mulțimea literelor inclusă în mulțimea literelor sale.

## 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

Alice și Bob comunică folosind următorul algoritm pentru a-și codifica mesajele:

- Alice și Bob cunosc amândoi o cheie secretă sub forma unui șir de caractere care este o permutare a celor 26 de litere mici din alfabetul englez, astfel: literei 'a' îi corespunde prima literă din cheia secretă, literei 'b' îi corespunde a doua literă din cheia secretă, ..., literei 'z' îi corespunde ultima literă din cheia secretă, după cum se poate observa din următorul exemplu:

**Alfabetul englez** = a b c d e f g h i j k l m n o p q r s t u v w x y z

**Cheia secretă** = o b c g s e f h i z j k l m n p q r d t u v a w x y

- Codificarea unui mesaj presupune înlocuirea fiecărei litere din el cu litera corespunzătoare din cheia secretă. De exemplu, dacă Alice și Bob au cheia secretă "obcgsefhizjklmnpqrduvawxy", atunci cuvântul "astazi" se va codifica prin cuvântul "odtoyi", deoarece literei 'a' îi corespunde litera 'o', literei 's' îi corespunde litera 'd' ș.a.m.d.
- Decodificarea unui mesaj codificat presupune căutarea fiecărei litere din el în cheia secretă și înlocuirea ei cu litera corespunzătoare din alfabetul englez. De exemplu, dacă Alice și Bob au cheia secretă "obcgsefhizjklmnpqrduvawxy", atunci cuvântul codificat "endt" se va decodifica în cuvântul "fost", deoarece literei 'e' îi corespunde litera 'f', literei 'n' îi corespunde litera 'o' ș.a.m.d.

Mesajele sunt codificate și transmise cuvânt cu cuvânt. Toate literele din mesaje sunt litere mici din alfabetul englez și nu se folosesc semne de punctuație, cu excepția caracterului '-', care nu se criptează.

Eve este un hacker care interceptează traficul și reușește să găsească algoritmul de decodificare, precum și cheia secretă. Eve are dificultăți în reconstrucția propozițiilor deoarece nu salvează cuvintele în ordinea în care sunt trimise. Să se reconstruiască propozițiile trimise de Alice și Bob.

a) [1 p.] Fișierul text *comunicare.in* are următoarea structură:

- pe prima linie se găsește cheia secretă
- pe fiecare dintre următoarele linii se găsesc informațiile despre un cuvânt (**transmis la cel puțin un minut distanță**), despărțite printr-un spațiu astfel:
  - primul caracter este A dacă cuvântul este trimis de Alice sau B dacă este trimis de Bob
  - separat printr-un spațiu se va găsi cuvântul în formă codificată
  - ultima informație va fi ora la care este trimis mesajul în format de 5 caractere și 24 de ore (de exemplu, 12:34 sau 21:03)

Să se scrie o funcție *citire\_date* care să returneze o structură cu datele din fișier.

b) [1.5 p.] Să se scrie o funcție *decodificare* care primește ca parametri un cuvânt codificat și cheia secretă utilizată. Funcția trebuie să decodifice **eficient** cuvântul codificat și apoi să-l returneze.

**[1.5 p.]** Să se reconstituie propozițiile trimise de Bob și Alice astfel :

- se decodifică fiecare cuvânt folosind funcția definită la punctul b)
- se determină ordinea în care au fost trimise cuvintele
- se salvează în fișierul text *comunicare.out* propozițiile reconstituite, astfel:

comunicare.in	după decriptarea datelor	comunicare.out
obcgsefhizjklmnpqrdtuvawxy B cnlprnlido 13:20 A ko 12:00 A vnl 10:00 B o 10:20 A dug 12:30 B rstrofsti-vo 14:20 A odtoyi 09:00 B cnlumicorso 09:20 A otoco 11:00 B endt 12:20	B compromisa 13:20 A la 12:00 A vom 10:00 B a 10:20 A sud 12:30 B retrageti-va 14:20 A astazi 09:00 B comunicarea 09:20 A ataca 11:00 B fost 12:20	A : astazi vom ataca la sud B : comunicarea a fost compromisa retrageti-va