

TEST DE LABORATOR LA DISCIPLINA "PROGRAMAREA ALGORITMILOR" VARIANTA 1

Subiectul 1 – 4 p

Fișierul "**cuvinte.in**" are $n > 1$ linii care conțin fiecare, separate prin câte un spațiu, câte n cuvinte formate doar din litere mici ale alfabetului englez.

a) [0.5 p] Să se scrie o funcție fără parametri numită **citire_cuvinte** care să citească datele din fișier și să returneze un obiect de tip list reprezentând matricea de dimensiuni $n \times n$ care conține cuvintele în ordinea din fișier. Pentru fișierul din exemplu matricea returnată va fi [['aceasta', 'este', 'o', 'propoziție'], ['este', 'este', 'dar', 'este'], ['cu', 'cateva', 'cuvinte', 'doar'], ['o', 'o', 'o', 'o']]

b) [1p] Fie **C** matricea returnată de **citire_cuvinte**. Se citește de la tastatură un număr **k**. Să se determine mulțimea cuvintelor (elementelor) din **C** cu minim **k** vocale. Să se scrie în fișierul **vocale.out** aceste cuvinte ordonate descrescător lexicografic, câte unul pe linie. Dacă nu există cuvinte cu proprietatea cerută se va scrie în fișier mesajul "Imposibil!".

c) [2p] Să se scrie o funcție **insereaza_inainte** care primește 3 parametri:

- o matrice de cuvinte **matrice_cuvinte** (de tip list)
- un șir de caractere **x**
- un șir de caractere **y**

Funcția modifică matricea **matrice_cuvinte** astfel:

- pe fiecare linie din matrice se inserează înainte de fiecare cuvânt **x** un nou element cu valoarea **y**, apoi
- se determină **k** = lungimea maximă a unei linii după inserări
- se completează fiecare linie a matricei cu cuvântul "." până ajunge de lungime **k** (se inserează la finalul fiecărei linii cuvântul "." până când linia devine de lungime **k**)

d) [0.5 p] Să se apeleze funcția **insereaza_inainte** pentru matricea returnată de **citire_cuvinte** și cuvintele 'este' și 'nu' și să se afișeze pe ecran matricea astfel obținută. Cuvintele de pe o linie a matricei se vor afișa pe o linie, separate cu spațiu.

cuvinte.in	Punctul b) k=2 vocale.out	Punctul d) ecran
<p>0 1 2 3</p> <p>aceasta este o propoziție este este dar este cu cateva cuvinte doar o o o o</p>	<p>propoziție este doar cuvinte cateva aceasta</p>	<p>0 1 2 3 4</p> <p>aceasta nu este o propoziție . . nu este nu este dar nu este cu cateva cuvinte doar . . . o o o o . . .</p>

Subiectul 2 – 5 p

O mare corporație a scos la concurs mai multe posturi. Pentru fiecare post concursul a avut două probe (numite **Proba 1** și **Proba 2**), fiecare cu punctaj de la **0** la **100**. Fișierul "**concurs.in**" conține rezultatele acestui concurs, sub următoarea formă: rezultatele la concursul pentru un post încep cu o linie de forma

Post: numele_postului

apoi urmează mai multe linii cu rezultatele candidaților de forma

Nume_candidat nume_proba punctaj

unde **Nume_candidat** este numele candidatului, **nume_proba** numele probei (poate fi doar Proba 1 sau Proba 2), iar **punctaj** este punctajul obținut de candidat la această probă.

concurs.in		
Post: Programator 1		
Marinescu Marian	Proba 1	100
Ionescu Matei	Proba 2	80
Albu Marius	Proba 1	10
Ionescu Matei	Proba 1	90
Marinescu Marian	Proba 2	50
Albu Marius	Proba 2	10
Post: Testare jocuri		
Pascu Petre	Proba 2	100
Marinescu Marian	Proba 1	80
Pascu Petre	Proba 1	100
Ionescu Matei	Proba 1	80
Ionescu Matei	Proba 2	9
Anton Alin	Proba 2	70
Marinescu Marian	Proba 2	80
Anton Alin	Proba 1	90
Post: Dezvoltator web		
Antonie Radu	Proba 1	100
Antonie Radu	Proba 2	100

a) [2,5 p.] Să se memoreze datele din fișier într-o singură structură de date astfel încât să se răspundă cât mai eficient la cerințele de la punctele următoare.

b) [1,5 p.] Scrieți o funcție **candidati_eligibili** care primește următorii parametri (în această ordine):

- structura în care s-au memorat datele la cerința a)
- un număr variabil de șiruri de caractere reprezentând numele unor posturi scoase la concurs
- un parametru *suma_minima* cu valoare implicită 50
- un parametru *punctaj_minim* cu valoare implicită 10.

Spunem că un candidat este eligibil pentru a ocupa un post la concurs pentru care a concurat dacă are punctajul la fiecare probă mai mare sau egal cu *punctaj_minim* și punctajul total (egal cu suma punctajelor la cele două probe) mai mare sau egal cu *suma_minima*.

Funcția **candidati_eligibili** returnează o listă de tupluri (**nume_post**, **nume_candidat**, **lista_punctaje**) cu candidații eligibili pentru un posturile primite ca parametru, unde:

- *nume_post* este un nume de post scos la concurs dintre cele primite ca parametru
- *nume_candidat* este un nume de candidat care a participat la concursul pentru postul *nume_post*
- *lista_punctaje* este o listă cu două elemente, reprezentând punctajul obținut de candidatul *nume_candidat* la Proba 1 și Proba 2 pentru postul *nume_post*

Lista returnată va fi ordonată crescător după numele postului, apoi, în caz de egalitate, descrescător după punctajul total al candidaților, iar în caz de egalitate crescător după nume candidat. **Explicații:** Pentru posturile 'Programator 1', 'Testare jocuri', dacă nu se dau valori la apel pentru parametrii *suma_minima* și *punctaj_minim* (deci se folosesc valorile implicite), lista returnată va fi [('Programator 1', 'Ionescu Matei', [90, 80]), ('Programator 1', 'Marinescu Marian', [100, 50]), ('Testare jocuri', 'Pascu Petre', [100, 100]), ('Testare jocuri', 'Anton Alin', [90, 70]), ('Testare jocuri', 'Marinescu Marian', [80, 80])]

c) [1 p.] Scrieți o funcție **actualizare_punctaje** care primește următorii parametri (în această ordine):

- structura în care s-au memorat datele la cerința a)
- o listă *punctaje_noi* de tuple-uri cu elemente de forma (*nume_candidat*, *punctaj*)
- un șir de caractere *nume_post*
- un număr *numar_proba* din mulțimea {1,2} reprezentând numărul probei de concurs

Pentru fiecare element (*nume_candidat*, *punctaj*) din lista *punctaje_noi* se va actualiza punctajul candidatului *nume_candidat* la proba cu numărul *numar_proba* pentru postul *nume_post* cu valoarea *punctaj* (doar dacă acesta a participat la probă, deci se află în structura de date memorată ca parametru).

Să se apeleze funcția pentru un nume de candidat, un punctaj, un nume de post și un număr de probă citite de la tastatură (fiecare dată de intrare se dă pe o linie). După apel să se afișeze structura în care s-au memorat datele.

De exemplu, pentru datele de intrare

Ionescu Matei

100

Programator 1

1

punctajul lui Ionescu Matei la Proba 1 a concursului pentru postul Programator 1 va deveni 100

MODALITATEA DE DESFĂȘURARE A TESTULUI DE LABORATOR LA DISCIPLINA "PROGRAMAREA ALGORITMILOR"

- Testul de laborator la disciplina "Programarea algoritmilor" se va desfășura în ziua de **13.01.2024**, în două runde, între orele 9⁰⁰ și 11⁰⁰, respectiv 11³⁰ și 13³⁰, astfel:
 - **Prima rundă**
 - 09⁰⁰ – 09¹⁵: efectuarea prezenței studenților
 - 09¹⁵ – 10⁴⁵: desfășurarea testului
 - 10⁴⁵ – 11⁰⁰: trimiterea surselor folosind un formular Google dedicat
 - **A doua rundă**
 - 11³⁰ – 11⁴⁵: efectuarea prezenței studenților
 - 11⁴⁵ – 13¹⁵: desfășurarea testului
 - 13¹⁵ – 13³⁰: trimiterea surselor folosind un formular Google dedicat
- Testul de laborator se va desfășura în laboratoarele Facultății de Matematică și Informatică, folosind calculatoarele din ele.
- Calculatoarele din laboratoare vor conține documentația oficială a limbajului Python, în format offline.
- Pe parcursul testului este interzisă utilizarea Internet-ului sau a oricărei alte forme de comunicare/informare, cu excepția documentației offline.
- În momentul efectuării prezenței, fiecare student trebuie să prezinte buletinul sau cartea de identitate.
- Testul va conține **două 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 complet al studentului și grupa sa. Dacă un student nu reușește să rezolve deloc un anumit subiect, totuși el 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%20etica%20FMI.pdf)).