

TEST DE LABORATOR LA DISCIPLINA "PROGRAMAREA ALGORITMILOR"
Sesiunea de examene ianuarie 2025 – VARIANTA 1

Subiectul 1 – 2 p

Fișierul *matrice.in* conține elementele unei matrice (pe fiecare linie din fișier sunt elementele unei linii din matrice separate cu un spațiu), numere naturale cu cel mult 9 cifre. Să se memoreze matricea din fișier într-o variabilă, apoi să se șteargă de pe fiecare linie numerele care conțin cel puțin două cifre egale cu lungimea numărului. Dacă există linii rămase vide (din care s-au șters toate elementele), acestea trebuie eliminate din matrice. Să se scrie matricea obținută în fișierul *matrice.out* (pe fiecare linie din fișier se vor scrie elementele unei linii din matrice separate prin câte un spațiu).

Exemplu:

matrice.in	matrice.out
333 52 4342 123	52 123
303 4044 331 53855	56789 431 2536 10
22 330 4554 338	27 1758
56789 431 2536 10	
27 323 133 1758	

Subiectul 2 – 2 p

Un algoritm de compresie presupune codificarea reversibilă a datelor astfel încât acestea să fie stocate într-o zonă de memorie de dimensiune mai mică decât cea inițială și în urma procesului de decompresie să se obțină exact datele inițiale. Un algoritm simplu de compresie a unui șir de cifre constă în înlocuirea unei secvențe de forma

$$\underbrace{x, x, x, \dots, x}_{y \geq 2 \text{ ori}}$$

cu perechea (x, y) . Pe fiecare linie a fișierului text *comprimat.in* se află câte un șir de cifre codat folosind algoritmul de compresie descris anterior. Scrieți un program care realizează decompresia datelor și salvează rezultatele în fișierul *decomprimat.out*.

Exemplu:

comprimat.in	decomprimat.out
1, (2,3), 1, 4, (5,2)	1, 2, 2, 2, 1, 4, 5, 5
2, 3, 4, 5, 6	2, 3, 4, 5, 6
(1,7), 4, 5	1, 1, 1, 1, 1, 1, 4, 5
1, 2, (3,4), 5, 6, (1,7), (2,3)	1, 2, 3, 3, 3, 3, 5, 6, 1, 1, 1, 1, 1, 1, 1, 2, 2, 2

Subiectul 3 – 5 p

Fișierul *zboruri.in* conține informații despre zborurile operate de diferite companii aeriene într-o zi. Fiecare linie are structura:

nume_companie, **oras_plecare** -> **oras_sosire**, **distanța_km**, **ore_de_plecare**
și corespunde unei rute de zbor, unde:

- **nume_companie** este numele companiei aeriene care operează zborul,
- **oras_plecare** este orașul de plecare,
- **oras_sosire** este orașul de destinație,
- **distanța_km** este distanța dintre orașele de plecare și sosire, exprimată în kilometri (număr întreg),
- **ore_de_plecare** este o listă de ore (format hh:mm) la care pleacă zborurile, separate prin spațiu, ordonată crescător.

Un exemplu de astfel de fișier este:

zboruri.in

Wizz Air , Bucuresti -> Milano , 1600 , 08:00 14:30 19:00
Ryanair , Bucuresti -> Dublin , 2500 , 09:00 15:00 21:00
KLM , Bucuresti -> New York , 7000 , 08:00 18:00
Tarom , Timisoara -> Londra , 2100 , 06:30 13:00 20:00
Wizz Air , Cluj-Napoca -> Londra , 2000 , 07:00 12:30 18:00
KLM , Cluj-Napoca -> Paris , 1800 , 09:30 17:30
Ryanair , Cluj-Napoca -> Roma , 1400 , 10:00 16:00
Wizz Air , Iasi -> Viena , 1200 , 06:30 13:45
Ryanair , Timisoara -> Paris , 1800 , 06:00 12:30 19:00
Tarom , Bucuresti -> Amsterdam , 2150 , 08:45 14:45
Tarom , Iasi -> Frankfurt , 1700 , 07:15 11:30 18:30
KLM , Bucuresti -> Madrid , 2500 , 10:00 16:00 22:00

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 p.] Scrieți o funcție **sterge_ultimul_zbor** care are următorii parametri (în această ordine):

- structura în care s-au memorat datele la cerința a),
- un șir de caractere **companie** reprezentând numele unei companii aeriene,
- un număr variabil de șiruri de caractere **orase_plecare** reprezentând nume de orașe de plecare.

Pentru compania *companie*, pentru fiecare oraș *oras* din *orase_plecare*, să se șteargă ultima oră din lista de ore de plecare a fiecărei rute a companiei care pleacă din orașul *oras*, dacă lista de ore are cel puțin 3 ore de plecare.

Să se apeleze funcția *sterge_ultimul_zbor* pentru a șterge ultimul zbor al companiei Tarom pentru orașele de plecare București și Iași. După apelul funcției să se afișeze și structura în care s-au memorat datele.

Explicații: Pentru compania Tarom și lista de orașe de plecare București și Iași, se va șterge ora 18:30 de la ruta Iași -> Frankfurt deoarece există 3 zboruri programate, iar pentru ruta București -> Amsterdam nu se va modifica nimic deoarece sunt doar două zboruri programate.

c) [1,5 p.] Scrieți o funcție *statistici_companii* care să primească structura de date creată la cerința a) și să returneze o listă de tuple-uri de forma:

(*nume_companie*, *numar_zboruri*, *distanța_totală*)

unde:

- **nume_companie** este numele unei companii,
- **numar_zboruri**: numărul total de zboruri (se iau în considerare toate orele pentru fiecare rută),
- **distanța_totală**: suma distanțelor parcurse într-o zi de toate zborurile companiei (se iau în considerare toate orele pentru fiecare rută).

Lista returnată trebuie să fie sortată descrescător după **distanța_totală**, în caz de egalitate, descrescător după **numar_zboruri**, iar în caz de egalitate, alfabetic după **nume_companie**. Apelați funcția și afișați informațiile returnate de funcție formate pe ecran în următorul format:

Compania ... are în total ... zboruri având în total ... kilometri.

Pentru datele inițiale (fără ștergere) se va afișa:

Compania KLM are în total 7 zboruri având în total 25100 kilometri.

Compania Ryanair are în total 8 zboruri având în total 15700 kilometri.

Compania Tarom are în total 8 zboruri având în total 15700 kilometri.

Compania Wizz Air are în total 8 zboruri având în total 13200 kilometri.

TEST DE LABORATOR LA DISCIPLINA "PROGRAMAREA ALGORITMILOR"
Sesiunea de examene ianuarie 2025 – VARIANTA 2

Subiectul 1 – 2 p

Fișierul *matrice.in* conține elementele unei matrice (pe fiecare linie din fișier sunt elementele unei linii din matrice separate cu un spațiu), numere întregi nenule. Să se memoreze matricea din fișier într-o variabilă, apoi pe fiecare linie a matricei fiecare număr negativ să fie înlocuit de două elemente noi, unul egal cu 0 și celălalt cu modulul numărului șters. Apoi să se șteargă din matrice liniile care nu conțin niciun 0. Să se scrie matricea obținută în fișierul *matrice.out* (pe fiecare linie din fișier se vor scrie elementele unei linii din matrice separate prin câte un spațiu).

Exemplu:

matrice.in	matrice.out
23 -14 -5 923	23 0 14 0 5 923
-11 2 -65 55	0 11 2 0 65 55
582 46 73 12	18 0 4 0 2 0 53
90 442 15 3	
18 -4 -2 -53	

Subiectul 2 – 2 p

Un algoritm de compresie presupune codificarea reversibilă a datelor astfel încât acestea să fie stocate într-o zonă de memorie de dimensiune mai mică decât cea inițială și în urma procesului de decompresie să se obțină exact datele inițiale. Un algoritm simplu de compresie a unui număr natural constă în înlocuirea unei secvențe de cifre de forma

$$\frac{xxx \dots x}{y \geq 2 \text{ ori}}$$

cu perechea (x, y) . Pe fiecare linie a fișierului text *decomprimat.in* se află câte un număr natural. Scrieți un program care realizează compresia numerelor folosind algoritmul descris mai sus și salvează rezultatele obținute în fișierul *comprimat.out*, câte unul pe linie.

Exemplu:

decomprimat.in	comprimat.out
12221455	1, (2, 3), 1, 4, (5, 2)
23456	2, 3, 4, 5, 6
111111145	(1, 7), 4, 5
12333356111111222	1, 2, (3, 4), 5, 6, (1, 7), (2, 3)
1223334444555556	1, (2, 2), (3, 3), (4, 4), (5, 5), 6

Subiectul 3 – 5 p

Fișierul `zboruri.in` conține informații despre zborurile operate de diferite companii aeriene într-o zi. Fiecare linie are structura:

`nume_companie`, `oras_plecare` -> `oras_sosire`, `distanța_km`, `ore_de_plecare`
și corespunde unei rute de zbor, unde:

- `nume_companie` este numele companiei aeriene care operează zborul,
- `oras_plecare` este orașul de plecare,
- `oras_sosire` este orașul de destinație,
- `distanța_km` este distanța dintre orașele de plecare și sosire, exprimată în kilometri (număr întreg),
- `ore_de_plecare` este o listă de ore (format hh:mm) la care pleacă zborurile, separate prin spațiu, ordonată crescător.

Un exemplu de astfel de fișier este:

`zboruri.in`

```
Wizz Air , Bucuresti -> Milano , 1600 , 08:00 14:30 19:00
Ryanair , Bucuresti -> Dublin , 2500 , 09:00 15:00 21:00
KLM , Bucuresti -> New York , 7000 , 08:00 18:00
Tarom , Timisoara -> Londra , 2100 , 06:30 13:00 20:00
Wizz Air , Cluj-Napoca -> Londra , 2000 , 07:00 12:30 18:00
KLM , Cluj-Napoca -> Paris , 1800 , 09:30 17:30
Ryanair , Cluj-Napoca -> Roma , 1400 , 10:00 16:00
Wizz Air , Iasi -> Viena , 1200 , 06:30 13:45
Ryanair , Timisoara -> Paris , 1800 , 06:00 12:30 19:00
Tarom , Bucuresti -> Amsterdam , 2150 , 08:45 14:45
Tarom , Iasi -> Frankfurt , 1700 , 07:15 11:30 18:30
KLM , Bucuresti -> Madrid , 2500 , 10:00 16:00 22:00
```

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 p.] Scrieți o funcție `sterge_ultimul_zbor` care are următorii parametri (în această ordine):

- structura în care s-au memorat datele la cerința a),
- un șir de caractere `companie` reprezentând numele unei companii aeriene,
- un număr variabil de șiruri de caractere `orase_plecare` reprezentând nume de orașe de plecare.

Pentru compania `companie`, pentru fiecare oraș `oras` din `orase_plecare`, să se șteargă ultima oră din lista de ore de plecare a fiecărei rute a companiei care pleacă din orașul `oras`, dacă lista de ore are cel puțin 3 ore de plecare.

c) [1,5 p.] Scrieți o funcție *statistici_companii* care să primească structura de date creată la cerința a) și să returneze o listă de tuple-uri de forma:

(*nume_companie*, *durata_medie_totala*, *numar_curse*)

unde:

- **nume_companie** este numele unei companii feroviare,
- **numar_curse**: numărul total de curse ale companiei (se iau în considerare toate orele pentru fiecare rută),
- **durata_medie_totala**: durata medie a curselor companiei, calculată ca media ponderată a duratelor medii pentru fiecare rută (cu 2 zecimale) - se iau în considerare toate orele pentru fiecare rută. Mai exact, **durata_medie_totala** pentru o companie se calculează astfel: pentru fiecare rută, se înmulțește durata medie a călătoriei cu numărul total de plecări pe acea rută; apoi se însumează toate aceste valori și se împart la numărul total de curse ale companiei

Lista returnată trebuie să fie sortată descrescător după **durata_medie_totala**, în caz de egalitate, crescător după **numar_curse**, iar în caz de egalitate, lexicografic (alfabetic) după **nume_companie**. Apelați funcția și afișați informațiile pe ecran formate în următorul format:

Compania ... are ... curse a caror durata medie este de ... minute.

Pentru datele inițiale (fără adăugarea de la b) se va afișa:

Compania CFR Calatori are 7 curse a caror durata medie este de 191.43 minute.

Compania RegioTrans are 9 curse a caror durata medie este de 176.67 minute.

Compania Transferoviar are 3 curse a caror durata medie este de 133.33 minute.