

Probleme FILE TXT

1.

O gospodină a înscris zilnic, o perioadă de timp, toate cheltuielile familiei, într-un fișier text *BUGET.TXT*. Pentru simplitate ea clasifica cheltuielile în 4 categorii:

- | | |
|------------------------|-------------------------|
| 1. Produse alimentare. | 3. Produse nealimentare |
| 2. Servicii comunale. | 4. Distracții. |

Fiecare linie din fișier conține numărul categoriei și valoarea cheltuielii.

Elaborați un program care citește datele din fișierul *BUGET.TXT*, calculează totalurile cheltuielilor pe fiecare categorie în parte și înregistre rezultatele într-un alt fișier *OUT.TXT*.

Intrare: Fișierul text *BUGET.TXT* conține un șir de linii. În fiecare linie sînt înscrise cîte două numere separate prin spațiu: numărul categoriei – un număr egal cu 1, 2, 3 sau 4 și suma cheltuielii – un număr real.

Ieșire: Fișierul text *OUT.TXT* conține patru linii. În fiecare linie, la începutul acesteia, se înregistrează denumirea categoriei, apoi, după un spațiu – totalul cheltuielilor la această categorie.

Exemplu	
BUGET.TXT	OUT.TXT
2 200	Produse alimentare 380
3 150	Servicii comunale 700
1 250	Produse nealimentare 150
4 120	Distracții 120
1 130	
2 500	

2.

În N localități pe parcursul a M zile a fost înregistrată la aceeași oră temperatura aerului. Localitățile sunt numerotate cu indicii: $1, 2, \dots, N$. Datele înregistrărilor formează tabloul A cu N linii și M coloane. Elementul $A[i, j]$ indică temperatura înregistrată în localitatea cu indicele i în ziua j .

Sarcină: scrieți un program, care va determina diferența dintre temperatura maximă și cea minimă înregistrate în fiecare localitate.

Pentru determinarea diferenței maxime de temperaturii în localitatea i , veți scrie în cadrul programului o funcție cu numele DELTA, care va primi în calitate de parametru valoarea întreagă i – indicele localității. Funcția va returna o valoare întreagă – diferența dintre valoarea maximă și valoarea minimă din linia i a tabloului.

Intrare: Fișierul text *VREME.IN* conține pe prima linie două numere întregi, separate prin spațiu: N - numărul de localități și M - numărul de zile în care s-a înregistrat temperatura ($1 \leq N \leq 10, 1 \leq M \leq 30$). Urmează N linii care conțin câte M valori numerice întregi, separate prin spațiu – elementele tabloului A . Numărul cu indicele j din linia $i+1$ este valoarea elementului $A[i, j]$.

Ieșire: Fișierul text *VREME.OUT* conține o singură linie, în care se vor înscrive N numere întregi: diferența maximă de temperaturi pentru fiecare din cele N localități.

Exemplu:

Vreme.in	Vreme.out	Explicație
3 4	7 1 10	Diferența dintre cel mai mare și cel mai mic element pe linii: linia 1: $12 - 5 = 7$; linia 2: $9 - 8 = 1$; linia 3: $8 - (-2) = 10$.
5 6 12 6		
8 9 9 8		
7 -2 8 7		

Rezolvarea va fi apreciată pentru:
declararea tipurilor de date și a variabilelor; operarea cu fișierele text; citirea și scrierea datelor; organizarea algoritmilor.

3. Agenția Națională pentru Reglementare în Energetică (ANRE) monitorizează importul de combustibil pe piața din Republica Moldova. În particular, datele despre importul zilnic de gaze (în mii m³) se înscriu într-un fișier text. Datele din fișier se înscriu în tabloul unidimensional **A**, în care valoarea elementului **A[i]** corespunde volumului de gaze, importat în ziua **i** a anului.

Sarcină: scrieți un program, care va citi din fișierul text **gaze.in** datele despre importul de gaze și va determina valoarea maximă a importului pe un interval dat de zile a anului.

Intrare: fișierul text **gaze.in** conține pe prima linie un număr întreg **N** ($1 \leq N \leq 365$) – numărul de zile trecute de la începutul anului (numărul de elemente în tabloul **A**). Următoarea linie conține **N** numere întregi, separate prin spațiu: importurile zilnice de gaze, în ordinea numerotării zilelor anului (elementele tabloului **A**, în ordinea creșterii indicilor). Ultima linie conține două numere întregi, separate prin spațiu: **s**, **f** – indicii zilelor ce corespund începutului și respectiv sfârșitului intervalului în care se cere să se determine importul zilnic maxim.

Ieșire: la ecran se va afișa un singur număr: volumul maxim de gaze importate zilnic în perioada dintre zilele cu indicii **s** și **f**, inclusiv.

Exemplu:

gaze.in	Ecran	Explicații	Rezolvarea va fi apreciată pentru: declararea tipurilor de date și a variabilelor; operarea cu fișierele text; citirea și scrierea datelor; utilizarea algoritmilor.
5 12 6 8 11 9 2 4	11	De la elementul cu indicele 2 și până la elementul cu indicele 4, valoarea maximă este 11.	

4. O agenție de turism gestionează **n** trasee turistice, numerotate de la 1 la **n**. Pentru a determina cel mai popular traseu au fost analizate ultimele **k** excursii de pe fiecare traseu. Cel mai popular este considerat traseul, pentru care numărul total de turiști în aceste **k** excursii este maximal.

Sarcină: Elaborați un program care determină cel mai popular traseu turistic. Programul trebuie să conțină un subprogram cu numele **TR**, care primind de la programul apelant în calitate de parametru un număr întreg – indicele traseului, returnează numărul total de turiști, care au vizitat traseul dat în ultimele **k** excursii.

Intrare: Fișierul text **turist.in** conține în prima linie două numere întregi - numărul de trasee turistice **n** ($1 \leq n \leq 8$) și numărul de excursii **k** ($1 \leq k \leq 10$). Fiecare din următoarele **n** linii conțin câte **k** numere întregi separate prin spațiu. Linia **i+1** conține numărul de turiști pentru fiecare din ultimele **k** excursii de pe traseul **i**.

Ieșire: Fișierul text **turist.out** conține un număr întreg – indicele celui mai popular traseu turistic.

Notă. Nu sunt două trasee turistice cu același număr total de turiști în ultimele **k** excursii.

Exemplu:

turist.in	turist.out	Rezolvarea va fi apreciată pentru: descrierea tipurilor de date și declararea variabilelor; operarea cu fișierele text; citirea și scrierea datelor; organizarea algoritmilor.
3 4 25 21 15 30 18 35 22 28 14 19 24 20	2	

5.

Pompele digitale din stația PECO înregistrează cantitatea de combustibil, în litri, solicitată pentru alimentarea automobilului. În decurs de o zi au fost efectuate N alimentări, iar datele au fost stocate în fișierul **PETROL.TXT**.

Sarcină: Scrieți un program care determină dacă volumul total al combustibilului solicitat pentru alimentare în stația PECO depășește cantitatea de 3000 litri.

Intrare: Fișierul text **PETROL.TXT** conține în prima linie un număr întreg N ($0 < N \leq 100$) – numărul de alimentări efectuate. Următoarea linie conține N numere întregi, separate prin spațiu – cantitățile solicitate de combustibil.

Ieșire: La ecran se va afișa în prima linie cantitatea totală de combustibil solicitată. Linia a doua va conține mesajul DA, dacă cantitatea totală de combustibil solicitată depășește 3000 sau mesajul NU, în caz contrar.

Exemplu:

PETROL.TXT:	Ecran:
6 30 50 20 60 10 35	205 NU
Rezolvarea va fi apreciată pentru: definirea tipurilor de date și declararea variabilelor; operarea cu fișierul text; citirea și scrierea datelor; implementarea algoritmului.	

6.

Concursurile la științe (matematică, fizică și informatică) se desfășoară în două etape. La prima etapă participanții sunt evaluați separat pe discipline. Pentru a fi admis la etapa a doua, elevul trebuie să acumuleze la fiecare dintre cele trei discipline cel puțin câte un punct. Suplimentar, suma punctelor acumulate trebuie să fie mai mare sau egală cu un număr prestabilit P .

Sarcină: Scrieți un program care numără câți elevi sunt admiși la etapa a doua a concursului. Programul va conține un subprogram cu numele **SM**, care primind în calitate de parametri numerele întregi **a**, **b**, **c** – punctele acumulate de un elev în prima etapă la matematică, fizică și informatică, va returna în programul apelant suma valorilor **a**, **b**, **c**.

Intrare: Fișierul text **concurs.in** conține în prima linie două numere întregi separate prin spațiu: **N** ($1 \leq N \leq 30$) – numărul de participanți la concurs și **P** – numărul minim de puncte, necesar pentru admiterea la etapa a doua. Fiecare din următoarele **N** linii conține trei numere întregi separate prin spațiu - punctele acumulate de un elev distinct în prima etapă, pe discipline aparte.

Ieșire: Fișierul text **concurs.out** conține un număr întreg – numărul de elevi admiși la etapa a doua a concursului.

Exemplu:

concurs.in	concurs.out	Rezolvarea va fi apreciată pentru:
4 15 10 9 0 5 4 8 5 5 3 8 7 10	2	descrierea tipurilor de date și declararea variabilelor; operarea cu fișierele text; citirea și scrierea datelor; organizarea algoritmilor.

7.

N bicicliști au participat la o cursă de ciclism cu **M** etape. Pentru ciclistul (**i**) viteza medie de parcurgere a etapei (**j**) este înregistrată în elementul **T[i, j]** al tabloului bidimensional **T**.

Sarcină: Scrieți un program, care determină pentru fiecare ciclist cea mai înaltă viteză medie pe etapă atinsă de acesta, după toate etapele cursei. Programul va conține o funcție cu numele **VM**, care primește în calitate de parametru valoarea întreagă **i** – numărul ciclistului. Funcția va returna un număr real – cea mai mare viteză medie a ciclistului **i** pentru toate etapele cursei.

Intrare: Fișierul text *cursa.in* conține pe prima linie două numere întregi **N** și **M** - numărul de cicliști și de etape ale concursului, separate prin spațiu. Urmează **N** ($1 \leq N \leq 10$) linii, care conțin câte **M** ($1 \leq M \leq 10$) numere reale, separate prin spațiu – elementele tabloului **T**. Numărul cu indicele **j** din linia **i+1** a fișierului este valoarea elementului **T[i, j]**.

Ieșire: Fișierul text *cursa.out* va avea **N** linii. Linia **i** va conține un număr real – cea mai înaltă viteză medie pe etapă pentru ciclistul **i**.

Exemplu:

<i>cursa.in</i>	<i>cursa.out</i>
4 3	65.3
65.3 43.2 48.5	61.3
48.0 52.8 61.3	77.0
77.0 62.8 39.6	68.1
68.0 54.0 68.1	

Rezolvarea va fi apreciată pentru:

descrierea tipurilor de date și declararea variabilelor; operarea cu fișierele text; citirea și scrierea datelor; organizarea algoritmilor.

8.

Lifturile moderne sunt dotate cu dispozitive de înregistrare a deplasării între etaje și a masei transportate, în kilograme. Fiecare deplasare este înregistrată printr-o pereche de numere, marcând direcția (1 – în sus, 2 – în jos) și masa transportată. Datele se acumulează în fișierul **LIFT.TXT**.

Sarcină: Scrieți un program, care afișează pe ecran unul din cuvintele:

- UP – dacă masa totală transportată în sus depășește masa totală transportată în jos;
- DOWN – dacă masa totală transportată în jos depășește masa totală transportată în sus;
- EQUAL – dacă masa totală transportată în jos este egală cu masa totală transportată în sus.

Intrare: Fișierul text **LIFT.TXT** conține în prima linie un număr întreg **N** ($0 < N < 100$) – numărul de înregistrări a deplasărilor liftului. Fiecare din următoarele **N** linii conține descrierea unei deplasări între etaje: cifra 1 sau 2, apoi, după spațiu – masa transportată.

Ieșire: La ecran se va afișa unul din cuvintele: UP, DOWN sau EQUAL – valoarea determinată din compararea maselor totale transportate în sus și în jos.

Exemplu:

LIFT.TXT:	Ecran:	Explicație:
6	DOWN	Suma greutateților transportată în sus (1): 100+0+200=300
1 100		
2 0		
2 80		Suma greutateților transportată în jos (2): 0+80+300=380
1 0		
1 200		
2 300		
Rezolvarea va fi apreciată pentru: definirea tipurilor de date și declararea variabilelor; operarea cu fișierul text; citirea și scrierea datelor; implementarea algoritmului.		