

PROBLEME PROPUSE

FIȘA NR. 1

Fișierul **intrare.txt** conține două linii. Pe prima linie este scris un număr natural nenul **n**, ($5 < n < 10$). Pe cea de-a doua linie a fișierului sunt scrise **n** numere întregi separate prin câte un spațiu, formate fiecare din cel mult 4 cifre, reprezentând un șir de **n** întregi.

Să se scrie un program în limbajul C++, care:

- să afișeze pe ecran, în linie, valorile absolute ale numerelor din șir, separate prin câte un spațiu;
- să afișeze pe ecran, în linie, numărul de divizori pozitivi proprii pentru fiecare număr din șir, separați prin câte un spațiu;
- să scrie în fișierul **iesire.txt**, pe prima linie, toate numerele impare din șir, separate prin câte un spațiu.

Exemplu

intrare.txt						Date de iesire:							
6						a)	12	14	7	15	27	5	
12	-14	-7	15	27	5	b)	4	2	0	2	2	0	
						Fișierul iesire.txt conține							
						c)	-7	15	27	5			

FIȘA NR. 2

Fișierul **intrare.txt** conține două linii. Pe prima linie este scris un număr natural nenul **n**, ($5 < n < 10$). Pe cea de-a doua linie a fișierului sunt scrise **n** numere întregi separate prin câte un spațiu, formate fiecare din cel mult 4 cifre, reprezentând un șir de **n** numere întregi.

Să se scrie un program în limbajul C++, care:

- să afișeze pe ecran, în linie, valorile absolute ale numerelor din șir, separate prin câte un spațiu;
- să afișeze pe ecran, în linie, suma divizorilor pozitivi proprii pentru fiecare număr din șir (numerele afișate se vor separa prin câte un spațiu);
- să scrie în fișierul **iesire.txt**, pe prima linie, toate numerele pare din șir, separate prin câte un spațiu.

Exemplu:

intrare.txt						Date de iesire:							
6						a)	12	14	7	15	27	5	
12	-14	-7	15	27	5	b)	15	9	0	8	12	0	
						Fișierul iesire.txt conține							
						c)	12	-14					

FIȘA NR. 3

Fișierul **intrare.txt** conține două linii. Pe prima linie este scris un număr natural nenul **n**, ($5 < n < 30$). Pe cea de-a doua linie a fișierului sunt scrise **n** numere naturale separate prin câte un spațiu, formate fiecare din cel mult 9 cifre, reprezentând un șir de **n** numere naturale.

Să se scrie un program în limbajul C++, care:

- să afișeze pe ecran, în linie, toate numerele din șir, separate prin câte un spațiu;
- să afișeze pe ecran, în linie, toate numerele din șir formate numai din cifre pare (dacă nu există astfel de numere în șir se va afișa mesajul **"NU EXISTĂ NUMERE NUMAI CU CIFRE PARE"**)
- să citească de la tastatură două numere naturale nenule **p1** și **p2** ($1 < p1 < p2 < n$), să afișeze numerele din șir situate între pozițiile **p1** și **p2** inclusiv, și să scrie noul șir în fișierul **ieșire.txt**, pe o linie, numerele separându-se prin câte un spațiu.

Exemplu: de la tastatură se citesc: **p1=2** și **p2=4**.

Exemplu:

intrare.txt						Date de ieșire:							
6						a)	1233	22	1785	56	15657	457	
1233	22	1785	56	15657	457	b)	22						
						Fișierul ieșire.txt conține							
						c)	22	1785	56				

FIȘA NR. 4

Fișierul **intrare.txt** conține două linii. Pe prima linie este scris un număr natural nenul **n**, ($5 < n < 30$). Pe cea de-a doua linie a fișierului sunt scrise **n** numere naturale separate prin câte un spațiu, formate fiecare din cel mult 4 cifre, reprezentând un șir de **n** numere naturale distincte.

Să se scrie un program în limbajul C++, care:

- să afișeze pe ecran, în linie, toate numerele din șir, separate prin câte un spațiu;
- să afișeze pe ecran, pe linii diferite, cel mai mic număr din șir și poziția acestuia;
- să scrie în fișierul **ieșire.txt**, pe o linie, separate prin câte un spațiu, toate numerele *perfecte* din șirul dat (dacă nu există astfel de numere, se va afișa mesajul **"NU EXISTĂ NUMERE PERFECTE"**). Un număr este *perfect* dacă este egal cu suma divizorilor lui pozitivi, exceptându-l pe el însuși, de exemplu: **6 = 1+2+3**.

Exemplu:

intrare.txt						Date de ieșire:							
6						a)	28	11	81	496	6	100	
28	11	81	496	6	100	b)	6	5					
						Fișierul ieșire.txt conține							
						c)	28	496	6				

FIȘA NR. 5

Fișierul **intrare.txt** conține două linii. Pe prima linie este scris un număr natural nenul **n**, ($5 < n < 30$). Pe cea de-a doua linie a fișierului sunt scrise **n** numere naturale separate prin câte un spațiu, formate fiecare din cel mult 4 cifre, reprezentând un șir de **n** numere naturale distincte.

Să se scrie un program în limbajul **C++**, care:

- să afișeze pe ecran, în linie, toate numerele din șir, separate prin câte un spațiu;
- să afișeze pe ecran, pe linii diferite, cel mai mare număr din șir și poziția acestuia;
- să scrie în fișierul **iesire.txt**, pe o linie, separate prin câte un spațiu, numerele *supraperfecte* din șirul dat (dacă nu există astfel de numere, se va afișa mesajul "**NU EXISTĂ NUMERE SUPRAPERFECTE**"). Un număr este *supraperfect* dacă este mai mic decât suma divizorilor lui pozitivi, exceptându-l pe el însuși, de exemplu: $12 < 1+2+3+4+6$.

Exemplu:

intrare.txt						Date de iesire:						
6						a)	22	12	121	20	18	13
22	12	121	20	18	13	b)	121	3				
						Fișierul iesire.txt conține						
						c)	12	20	18			

FIȘA NR. 6

Fișierul **intrare.txt** conține două linii. Pe prima linie este scris un număr natural nenul **n**, ($5 < n < 30$). Pe cea de-a doua linie a fișierului sunt scrise **n** numere naturale separate prin câte un spațiu, formate fiecare din cel mult 4 cifre, reprezentând un șir de **n** numere naturale. Șirul conține cel puțin două numere pare. Să se scrie un program în limbajul **C++**, care:

- să afișeze pe ecran, în linie, toate numerele din șir, separate prin câte un spațiu;
- să afișeze pe următoarea linie a ecranului, media aritmetică a tuturor numerelor pare din șir ;
- să scrie în fișierul **iesire.txt**, pe o linie, separate prin câte un spațiu, numerele de tip palindrom din șirul dat (dacă nu există astfel de numere, se va afișa mesajul "**NU EXISTĂ NUMERE PALINDROM**"). Un număr este palindrom dacă numărul citit de la stânga la dreapta este egal cu numărul citit de la dreapta la stânga; de exemplu: **33, 141, 2552**.

Exemplu:

intrare.txt						Date de iesire:						
6						a)	2552	56	32	444	46	1221
2552	56	32	444	46	1221	b)	626					
						Fișierul iesire.txt conține						
						c)	2552	444	1221			

FIȘA NR. 7

Fișierul **intrare.txt** conține două linii. Pe prima linie este scris un număr natural nenul **n**, ($5 < n < 30$). Pe cea de-a doua linie a fișierului sunt scrise **n** numere naturale separate prin câte un spațiu, formate fiecare din cel mult 4 cifre, reprezentând un șir de **n** numere naturale. Șirul conține cel puțin două numere impare. Să se scrie un program în limbajul **C++**, care:

- să afișeze pe ecran, în linie, toate numerele din șir, separate prin câte un spațiu;
- să afișeze pe ecran, în linie, numărul de cifre din care este format fiecare număr din șirul inițial, numerele din linie separându-se prin câte un spațiu;
- să scrie în fișierul **iesire.txt**, pe prima linie, suma tuturor numerelor impare din șir.

Exemplu:

intrare.txt						Date de iesire:						
6						a)	13	555	71	2	242	1001
13	555	71	2	242	1001	b)	2	3	2	1	3	4
						Fișierul iesire.txt conține						
						c)	1640					

FIȘA NR. 8

Fișierul **intrare.txt** conține două linii. Pe prima linie este scris un număr natural nenul **n**, ($5 < n < 30$). Pe cea de-a doua linie a fișierului sunt scrise **n** numere întregi separate prin câte un spațiu, formate fiecare din cel mult 4 cifre, reprezentând un șir de **n** numere întregi. Să se scrie un program în limbajul **C++**, care:

- să afișeze pe ecran, în linie toate numerele din șir, separate prin câte un spațiu;
- să afișeze pe ecran, în linie, separate prin câte un spațiu, toate numerele prime din șir (dacă nu există numere prime în șir, se va afișa pe ecran un mesaj corespunzător);
- să scrie în fișierul **iesire.txt**, pe prima linie, suma tuturor numerelor pozitive din șir (dacă nu există numere pozitive în șir se va scrie în fișierul **iesire.txt** un mesaj corespunzător).

Exemplu:

intrare.txt						Date de iesire:						
6						a)	1234	-555	41	-2	-242	11
1234	-555	41	-2	-242	11	b)	41	11				
						Fișierul iesire.txt conține						
						c)	1268					

FIȘA NR. 9

- 1) În fișierul **numere.txt** se află mai multe numere naturale, de cel mult 3 cifre fiecare, scrise pe un singur rând. Să se scrie un program care creează un alt fișier **iesire.txt** care să conțină exact aceleași numere din fișierul **numere.txt**, câte unul pe linie, întâi numere pare apoi numerele impare. Exemplu: fișierul **numere.txt** conține pe un rând numerele: 12 37 25 512 122 42 54 301, fișierul **iesire.txt** va conține pe câte o linie numerele: 12 512 122 42 54 37 25 301.
- 2) Fișierul text **numere.txt** conține pe mai multe rânduri cel mult 50000 de numere naturale din intervalul închis $[0, 99]$, numerele de pe același rând fiind separate prin câte un spațiu. Scrieți un program care afișează pe ecran, acele numere din fișier care au cel puțin **k** divizori, **k** fiind un număr natural, $k < 20$, citit de la tastatură. Numerele vor fi afișate pe ecran separate prin câte un spațiu. Exemplu: dacă fișierul conține numerele: **15 36 33 36 1 12 1 24 2**, iar pentru **k** se citește valoarea **5**, se vor afișa numerele **36 36 12 24**.
- 3) Fișierul text **numere.txt** conține cel mult **10000** de numere naturale din intervalul închis $[0, 999]$, aflate pe mai multe linii, numerele de pe aceeași linie fiind separate prin câte un spațiu. Scrieți un program care determină și afișează pe ecran cel mai mic număr prim **a** cu proprietatea $\max \leq a$, unde **max** este cea mai mare valoare din fișier. **Exemplu:** dacă fișierul conține numerele: **5 8 99 5 1 1 2 2** se va afișa **101**.
- 4) Fișierul text **numere.txt** conține un șir de cel mult **2008** numere naturale nenule, cu cel mult **4** cifre fiecare, pe mai multe rânduri, numerele de pe același rând fiind separate prin câte un spațiu. Scrieți un program care citește de la tastatură un număr natural **k** și afișează pe ecran cel mai mare număr din fișierul **bac.txt** care este mai mic sau egal cu numărul natural **k**, precum și numărul de apariții ale acestuia în fișier. Cele două valori vor fi afișate pe o linie a ecranului, separate printr-un spațiu. Dacă în fișier nu există nici un număr mai mic sau egal cu **k**, se va afișa doar valoarea **0**. **Exemplu:** dacă în fișier avem numerele **31 2 63 71 8 63 5 281** și numărul citit este **k=70**, atunci pe ecran se vor afișa numerele: **63 2**.
- 5) Fișierul text **numere.txt** conține cel puțin două și cel mult 1000 de numere naturale, distincte. Numerele sunt separate prin câte un spațiu și fiecare dintre ele are cel mult 9 cifre. Scrieți un program care determină cele mai mici două numere din fișier. Cele două numere vor fi afișate pe ecran, în ordine crescătoare, separate printr-un spațiu. **Exemplu:** dacă fișierul conține numerele: **512 6 860 3 12 300** se va afișa: **3 6**.

FIȘA NR. 10

- 1) Pe prima linie a fișierului **BAC.TXT** se găsește un număr natural **n** ($0 < n < 1000$), pe linia a doua se găsesc **n** numere naturale, ordonate crescător, cu cel mult 4 cifre fiecare, iar pe linia a treia un număr natural **a** ($a < 10000$).

Numerele de pe a doua linie a fișierului sunt separate prin câte un spațiu. Să se scrie un program care citește toate numerele din fișier și afișează pe ecran mesajul **DA** dacă numărul **a** se află printre cele **n** numere de pe linia a doua a fișierului, sau mesajul **NU** în caz contrar. **Exemplu:** dacă fișierul conține:

7

2 3 3 4 5 8 9

8

atunci se va afișa: **DA**

- 2) Un număr natural se numește palindrom dacă numărul citit de la stânga la dreapta este egal cu numărul citit de la dreapta la stânga. Scrieți un program care citește de pe prima linie a fișierului text **NUMERE.IN** un număr natural **n** ($0 < n < 101$), apoi de pe următoarea linie **n** numere naturale de cel mult nouă cifre fiecare. Programul scrie în fișierul text **NUMERE.OUT** numerele palindroame de pe a doua linie a fișierului **NUMERE.IN**.

Exemplu: dacă fișierul **NUMERE.IN** conține

6

125 66 458 64146 323 4589

atunci **NUMERE.OUT** va conține numerele: **66 64146 323**.

- 3) Fișierul text **NUMERE.IN** conține pe prima linie un număr natural **n** ($0 < n < 1000$), iar pe a doua linie, separate prin câte un spațiu, **n** numere naturale cu cel mult nouă cifre fiecare. Scrieți un program care citește de la tastatură o cifră **c**, și determină pentru câte dintre numerele de pe a doua linie a fișierului **NUMERE.IN**, cifra **c** reprezintă cifra maximă. Programul scrie pe primul rând al fișierul text **NUMERE.OUT** mesajul **DA** dacă există astfel de valori și pe al doilea rând numărul acestora, iar dacă nu există va scrie în fișier mesajul **NU**.

Exemplu: dacă cifra citită este **8**

NUMERE.IN conține

10

725 3695 423 0 7895 0 100 101 870 568

NUMERE.OUT va conține

DA

2

FIȘA NR. 11

- 1) Fișierul text **NUMERE.TXT** conține pe prima linie un număr natural n ($1 \leq n \leq 10000$) și pe a doua linie, n numere naturale cu cel mult 9 cifre fiecare, numere nu neapărat distincte. Aceste numere sunt dispuse în ordine crescătoare și separate între ele prin câte un spațiu. Scrieți un program care citește valorile din fișier și afișează pe ecran, cu un spațiu între ele, valoarea care apare de cele mai multe ori în fișier și de câte ori apare ea. Dacă există mai multe valori care apar de un număr maxim de ori, se va afișa cea mai mică dintre ele. Exemplu: dacă fișierul **NUMERE.TXT** are conținutul de mai jos,

8

711 711 711 11111 11111 11111 191111

231111 atunci programul va afișa pe ecran **711 3**.

- 2) Fișierul text **bac.txt** conține, pe o singură linie, cel puțin 2 și cel mult 100 de numere naturale nenule distincte de cel mult 4 cifre fiecare, numerele fiind separate prin câte un spațiu. Scrieți un program care citește toate numerele din fișier și scrie pe ecran ultima cifră a produsului celor mai mari două numere dintre cele citite. **Exemplu:** dacă fișierul **bac.txt** conține numerele: **1017 48 312 5742 162** atunci se va afișa: **4** (ultima cifră a produsului numerelor **1017, 5742**).

- 3) Fișierul text **numere.txt** conține, pe o singură linie, cel mult **1000** de numere naturale nenule de cel mult patru cifre fiecare, numerele fiind separate prin câte un spațiu. Scrieți un program care citește toate numerele din fișier și scrie în fișierul text **pare.txt** toate numerele pare citite, în ordinea în care acestea apar în fișierul **numere.txt**. Dacă fișierul **numere.txt** nu conține niciun număr par, atunci în fișierul **pare.txt** se va scrie mesajul **nu exista**.

Exemplu: **numere.txt** 2 3 1 4 7 2 5 8 6, **pare.txt** 2 4 2 8 6.

- 4) Fișierul text **bac.txt** conține pe prima linie un număr natural n ($n < 100$), iar pe a doua linie, separate prin câte un spațiu, n numere naturale, mai mici decât **30000** fiecare. Scrieți un program care citește de la tastatură un număr natural k ($k < 10$) precum și numerele din fișierul **bac.txt** și determină și afișează pe ecran, cu câte un spațiu între ele, toate numerele de pe a doua linie a fișierului care conțin în scrierea lor cifra memorată în variabila k . Dacă nu există un asemenea număr pe a doua linie a fișierului, se afișează pe ecran mesajul **NU**. **Exemplu:** dacă se citește de la tastatură $k=2$, iar fișierul conține numerele

8

234 5678 317 809 202 427 92 6004, atunci pe ecran se afișează numerele: **234 202 427 92**.

FIȘA NR. 12

- 1) Fișierul **bac.txt** conține pe prima linie un număr natural **n** ($n \leq 100$), iar pe a doua linie, separate prin câte un spațiu, **n** numere naturale nenule, cu cel mult 4 cifre fiecare. Scrieți programul care citește de la tastatură un număr natural **k** ($k \leq 25$), și afișează pe ecran numerele citite din fișierul **bac.txt** care au cel puțin **k** divizori. Exemplu: dacă **k=9**, iar fișierul are conținutul
8
100 56 22 909 222 400 180 616,
atunci numerele care se afișează sunt: **100 400 180 616**.
- 2) Fișierul text **numere.txt** conține pe prima linie un număr natural **n** ($0 < n < 100000$) iar pe a doua linie, separate prin câte un spațiu, **n** numere naturale formate din cel mult 2 cifre. Scrieți un program care determină dacă numerele situate pe a doua linie a fișierului sunt în ordine strict crescătoare. În caz afirmativ, programul va afișa pe ecran mesajul **DA**, altfel va afișa mesajul **NU**. Exemplu: dacă fișierul **numere.txt** are următorul conținut:
7
3 5 2 1 5 23 1, atunci pe ecran se va afișa: **NU**.
- 3) Fișierul text **numere.txt** conține pe prima linie un număr natural **n** ($0 < n < 100000$), iar pe a doua linie **n** numere naturale de o singură cifră, separate prin câte un spațiu. Scrieți un program care determină cea mai mare și cea mai mică cifră dintre cele situate pe a doua linie a fișierului. Cifrele determinate vor fi afișate pe ecran, pe o singură linie, separate printr-un spațiu. Exemplu: dacă fișierul **numere.txt** are următorul conținut:
7
3 5 2 1 5 3 1, atunci pe ecran se vor afișa valorile: **5 1**.
- 4) Pe prima linie a fișierului **bac.in** se află un număr natural nenul **n** ($n \leq 1000$), iar pe a doua linie a fișierului se află un șir format din **n** numere naturale, despărțite prin câte un spațiu, fiecare număr fiind format din cel mult 4 cifre. Scrieți un program care citește valorile din fișier și care afișează pe ecran mesajul **DA** dacă toate elementele din șir sunt egale și mesajul **NU** în caz contrar. Exemplu: dacă fișierul **bac.in** are conținutul
7
10 10 10 10 10 10 10, pe ecran se va afișa: **DA**