

FIȘA 5. Fișiere text

Pentru fiecare dintre următoarele enunțuri scrieți programul C++ corespunzător.

1. Fie x_1, x_2, \dots, x_n un șir format din n ($n \geq 1$) numere naturale nenule distincte. Spunem că șirul x_1, x_2, \dots, x_n este o creastă dacă există un indice p ($1 < p < n$) astfel încât $x_1 < x_2 < \dots < x_{p-1} < x_p$ și $x_p > x_{p+1} > \dots > x_{n-1} > x_n$, iar numărul aflat pe poziția p (adică x_p) se numește vârful crestei.

Fișierul text BAC.IN conține pe prima linie un număr natural n cuprins între 1 și 100, iar pe a doua linie conține n numere naturale nenule distincte mai mici sau egale decât 5000, despărțite prin spații. Scrieți un program care să creeze un fișier text BAC.OUT în care să scrie, pe prima linie, numărul 0 dacă cele n numere aflate pe a doua linie a fișierului BAC.IN nu formează o creastă sau numărul care reprezintă vârful crestei, în caz contrar.

Exemple:

BAC.IN	BAC.OUT	BAC.IN	BAC.OUT
6	26	6	0
7 12 17 26 15 3		7 12 17 10 15 3	

2. Fișierul text BAC.TXT conține 100 de numere naturale de cel mult 6 cifre fiecare, câte un număr pe fiecare linie a fișierului. Scrieți un program care:

- a) afișează pe ecran toate numerele din fișier, câte 5 pe fiecare linie, cu excepția ultimei linii care poate avea mai puțin de 5 numere, numerele de pe o linie fiind separate prin câte un spațiu;
- b) afișează pe ecran suma numerelor prime din fișier.

3. Prima linie a fișierului BAC.TXT conține două numere naturale m și n ($1 \leq n, m \leq 100$) iar următoarele m linii câte n numere întregi cu maxim 4 cifre fiecare, separate prin câte un spațiu. Se cere să se afișeze pe ecran câte din cele m linii din fișier sunt simetrice. Spunem că o linie a fișierului este simetrică dacă elementele egal depărtate de capetele liniei respective sunt egale (primul element de pe linie este egal cu ultimul element al liniei, al doilea cu penultimul etc.)

De exemplu dacă fișierul BAC.TXT are următorul conținut

```
3 5
15 3 0 3 15
5 9 1 9 5
6 8 20 8 3
```

atunci se va afișa pe ecran valoarea 2 (primele două din cele m linii sunt simetrice).

4. Se consideră două fișiere text F1.TXT și F2.TXT ce conțin numere reale dispuse pe mai multe linii și separate prin spații. Fișierul F1.TXT conține numai numere distincte, iar fișierul F2.TXT conține numai numere din fișierul F1.TXT, dar acestea se pot repeta.

Scrieți programul C/C++ care afișează pentru fiecare valoare din fișierul F1.TXT numărul său de apariții în fișierul F2.TXT. Pe linii diferite ale ecranului vor fi afișate câte două valori, separate prin spațiu, prima fiind numărul din fișierul F1.TXT, iar a doua numărul său de apariții din F2.TXT.

5. Scrieți programul C/C++ care, pentru o valoare n (număr natural, $1 \leq n \leq 100$) citită de la tastatură, scrie în fișierul bac.txt, separate prin spațiu, primele n numere strict pozitive divizibile cu 3.

De exemplu, pentru $n=4$, conținutul fișierului este : 3 6 9 12

6. Se citește un număr n natural, $2 < n < 10000$ de la tastatură. Să se scrie un program C/C++ care creează fișierul text BAC.TXT și scrie în el, pe aceeași linie, cu separare prin spațiu, primele n cifre din șirul 1, 2, 3, 4, 5, 6, 7, 8, 9, 1, 2, 3, ... Cifrele i cu $i < 9$ sunt consecutive, iar după fiecare cifră de 9 urmează cifra 1.

De exemplu, dacă pentru n se citește valoarea 14, atunci fișierul va conține, în această ordine, valorile: 1 2 3 4 5 6 7 8 9 1 2 3 4 5.

7. Pe prima linie a fișierului text **BAC.IN** se află un număr natural n ($0 < n \leq 1000$), iar pe a doua linie n numere reale pozitive, despărțite prin câte un spațiu. Scrieți programul C/C++ care citește datele din fișierul de intrare și scrie în fișierul text **BAC.OUT** pe o linie, despărțite prin câte un spațiu, numerele care au partea întreagă număr prim. Dacă nici unul din numere nu are partea întreagă număr prim atunci se va scrie în fișierul **BAC.OUT** mesajul **NU EXISTA**
- De exemplu, dacă **BAC.IN** conține
- ```
6
12.095 31.567 5.789 789.834 1234.923 2.345
```
- atunci fișierul **BAC.OUT** va conține
- ```
31.567 5.789 2.345
```
8. Se citește de la tastatură un număr natural nenul n care are cel mult 9 cifre. Să se afișeze în fișierul text **Date.out** numărul k , natural, astfel încât produsul $1*2*3*...*(k-1)*k$ să aibă o valoare cât mai apropiată de numărul n .
- De exemplu, dacă se citește numărul $n=25$ fișierul **Date.out** are următorul conținut : 4
De exemplu dacă se citește numărul $n=119$ fișierul **Date.out** are următorul conținut : 5
9. Se citește de la tastatură un număr natural n ($0 < n < 100$). Scrieți programul C/C++ ce construiește fișierul **prime.out** cu primele n numere prime, în ordine crescătoare, separate prin câte un spațiu.
- Exemplu: Pentru $n=8$, fișierul **prime.out** conține: 2 3 5 7 11 13 17 19
10. Scrieți programul C/C++ care citește de la tastatură un număr natural n ($0 < n < 100$) și creează fișierul text **BAC.TXT** care conține pe prima linie, în ordine descrescătoare, toate numerele naturale de la n la 1, pe a doua linie în ordine descrescătoare, toate numerele naturale de la $n-1$ la 1 etc... pe linia $n-1$ numerele 2 1, iar pe ultima linie numărul 1. Pe fiecare linie numerele vor fi despărțite prin câte un spațiu.
- Exemplu: dacă se citește $n=3$, atunci **BAC.TXT** va conține:
- ```
3 2 1
2 1
1
```
11. Se citește de la tastatură un număr natural nenul  $n$  ( $n < 1000$ ). Scrieți programul C/C++ care construiește fișierul text **bac.txt** care să conțină, pe prima linie, toți divizorii lui  $n$  în ordine strict descrescătoare. Divizorii vor fi separați prin spațiu.
- De exemplu, dacă  $n=10$ , atunci fișierul **bac.txt** va conține:
- ```
10 5 2 1
```
12. Se consideră fișierul text **date.in** care conține exact 899 de numere distincte formate fiecare din câte 3 cifre. Să se afișeze în fișierul text **date.out** numărul format din 3 cifre care lipsește din fișierul text **date.in**.
13. Scrieți un program C/C++ care citește de pe prima linie a fișierului text **BAC.TXT** trei numere naturale nenule a, b, c formate din cel mult patru cifre fiecare, separate prin câte un spațiu și afișează pe ecran cel mai mare divizor comun al acestora. De exemplu, dacă din fișier se citesc numerele: 9 27 15, atunci se afișează 3.
14. Scrieți un program C/C++ care citește din fișierul text **DATE.IN**, cel mult 100 de numere naturale nenule aflate pe o singură linie, formate din cel mult patru cifre fiecare, separate prin spații și scrie aceste numere în fișierul text **DATE.OUT**, în ordine inversă față de cea în care fost citite, pe o singură linie, separate prin spații. De exemplu, dacă din fișierul **DATE.IN** se citesc numerele: 93 207 15 1892 3762, atunci conținutul fișierului **DATE.OUT** va fi: 3762 1892 15 207 93
15. Se citește de la tastatură un număr natural n de cel mult 8 cifre. Să se creeze fișierul text **NR.TXT** care să conțină, câte unul pe linie, în orice ordine, toate numerele naturale distincte care se pot obține din valoarea lui n prin eliminarea uneia sau mai multor cifre de la unul din capetele sale.
- De exemplu, pentru $n=38604$, fișierul **NR.TXT** va conține, câte unul pe linie și nu neapărat în această ordine: 8604 604 4 3860 386 38 3