Fisiere text

Teorie

- Fisierul este o colectie de date omogene memorate pe un suport extern. Fisierul se identifica printr-un nume si o extensie. Intr-un fisier text datele sunt memorate sub forma unei succesiuni de caracterere(memorate prin utiliarea codului ASCII). Fisierul text este format dn una sau mai multe linii, o linie se termina cu caracterul newline, mai putin ultima parte care se termina cu marcajul de sfarsit de fisier(CRTL+Z).

Lucrul cu fisiere text

- Pentru a lucra cu fisiere text trebuie sa includem header-ul fstream.

```
#include<fstream>
```

- Fiecarui fisier fizic ii corespunde in program un fisier logic(o variabila), care trebuie declarata inaintea utilizarii ei. Daca un fisier text este utilizat numai pentru operatii de citire acesta poate fi deschis astfel:

```
ifstream nume_logic("nume_fizic");
```

- Daca un fisier test este utilizat numai pentru operatii de scriere acesta poate fi deschis astfel:

```
ofstream nume_logic("nume_fizic");
```

Citirea si afisarea datelor

- Se considera declararile urmatoare:

```
ifstream fin("bac.in");
ofstream fout("bac.out");
int x;
// citirea variabilei x dn fisier
fin>> x;
// afisarea variabilei x in fisierul de iesire
fout<< x;
```

Citirea datelor pana la sfarsitul fisierului

- Presupunem ca avem enuntul: `Fisierul bac.txt contine valori intregi, acestea trebuie prelucrate, secventa<prelucreaza x> realizeaza prelucrarile dorite, aceasta va fi`

modificata conform cerintelor. Prelucrarea datelor intr-un fisier cand nu se cunoaste numarul de valori din fisier se poate face dupa un algoritm similar cu unul dintre algoritmii urmatori.

- Primul algoritm

```
ifstream fin("bac.txt");
int x;
while(fin>>x)
{
    <prelucreaza x>
}
```

- Al doilea algoritm

```
ifstream fin("bac.txt");
int x;
fin>>x
while(!fin.eof())
{
    <prelucreaza x>
    fin>>x;
}
```

- Testarea sfarsitului de fisier se poate face folosind functia eof() care nu citeste, ci doar testeaza daca anterior a fost detectat sfarsitul de fisier. Apelul functiei se face astfel:

```
nume_logic.eof();
```

- Inchiderea fisierelor text

```
nume_logic.close();
```

UNDE nume_logic este numele variabilei de tipul ifstrem/ofstream

Exercitii propuse

1. Fisierul text bac.txt contine, pe o singura linie, cel mult 1000 de numere naturale nenule, numerele fiind separate prin cate un spatiu. Scrieti un program C/C++ care citeste de la tastatura un numar natural nenul n si numerele din fisierul bac.txt si care afiseaza pe ecran, separate prin cate un spatiu, toate numerele din fisier care sunt divizibile cu n. Daca fisierul nu contine niciun astfel de numar, atunci se va afisa pe ecran mesajul NU EXISTA.

- Exemplu: daca fisierul bac.txt contine numerele: 3 100 40 70 25 5 80 6 3798, pentru n=10 atunci pe ecran se va afisa: 100 40 70 80

```
#include <iostream>
#include <fstream>
#include <cmath>

using namespace std;
ifstream f("bac.txt");

int main()
{
    int n, x, cnt;
    cout << "Enter n: ";
    cin >> n;

    cnt = 0;
    while (f >> x){
        if (x % n == 0)
        {
            cout << x << " ";
            cnt++;
        }
    }
    if (cnt == 0)
    {
        cout << "NU EXISTA";
    }
}
```

2. Se considera fisierul bac.txt ce contine un sir crescator cu cel mult un milion de numere naturale de cel mult noua cifre fiecare, separate prin cate un spatiu. Sa se scrie un program C/C++ care, folosind un algoritm eficient din punct de vedere al memoriei utilizate si al timpului de executare, citeste din fisier toti termenii sirului si afiseaza pe ecran, pe o singura linie, fiecare termen distinct al sirului urmat de numarul de aparitii ale acestuia in sir. Valorile afisate sunt separate prin cate un spatiu.

- Exemplu: daca fisierul bac.txt are urmatorul continut: 1 1 1 5 5 5 5 9 9 11 20 20 20 programul va afisa: 1 3 5 4 9 2 11 1 20 3 deoarece 1 apare de 3 ori, 5 apare de 4 ori, etc.

```
#include <iostream>
#include <fstream>
#include <cmath>

using namespace std;

int main()
{
    ifstream f("bac.txt");
    int x, y, cnt;
```

```
;
    f >> x;
    cnt = 1;
    while (f >> y){
        if (x == y)
        {
            cnt++;
        }
        else
        {
            cout << x << " " << cnt << " ";
            x = y;
            cnt = 1;
        }

    }
    cout << x << " " << cnt << " ";
}
```