**Vectori de marcaje**

**V[x]=1, daca elementul x a fost citit din fisier (e marcat)**

**V[x]=0, altfel**

**P1**) Fisierul IN.TXT contine cel mult 1000000 numere naturale de cel mul 2 cifre fiecare. Sa se afiseze in fisierul OUT.TXT elementele din primul fisier in ordine **strict crescatoare**.

**Ex.**

IN.TXT

79 45 67 4 3 22 45 79 45 4 3 4 3 3 4

OUT.TXT

3 4 22 45 67 79

**int v[100], x, mx;**

ifstream f (“IN.TXT”);

ofstream g (“OUT.TXT”);

int main()

{ while (f>>x) // se construieste vectorul de marcaje in timpul citirii numerelor din fisier

{ **v[x]=1**;

// if (x>mx) mx=x;

}

// se afiseaza numerele marcate in ordine crescatoare

for (i=0; i<100; i++) // sau for (i=0; i<=mx; i++)

if (v[i]) g<<i<<” ”;

return 0;

}

**Obs**. Deoarece construirea vectorului de marcaje se face in timpul citirii numerelor din fisier si afisarea numerelor in ordine strict crescatoare se face prin parcurgerea numerelor marcate de maxim doua cifre, algoritmul este liniar, de complexitate O(n) -> eficient d.p.d.v al timpului de executare.

**Vectori de frecvente / aparitii**

**V[x] = numarul de aparitii al numarului x (in fisier)**

**P1**) Fisierul IN.TXT contine cel mult 1000000 numere naturale de cel mul 2 cifre fiecare. Sa se afiseze in fisierul OUT.TXT elementele **distincte** din primul fisier care apar de cel putin doua ori, in ordine **crescatoare**.

**Ex.**

IN.TXT

79 45 67 4 3 22 45 79 45 4 3 4 3 3 4

OUT.TXT

3 4 45 79

**int v[100], x, mx;**

ifstream f (“IN.TXT”);

ofstream g (“OUT.TXT”);

int main()

{ while (f>>x) // se construieste vectorul de frecvente in timpul citirii numerelor din fisier

{ **v[x]++;**

if (x>mx) mx=x;

}

// se afiseaza numerele care au cel putin frecventa 2

for (i=0; i<=mx; i++)

if (v[i]>=2) g<<i<<” ”;

return 0;

}

**P2**) Fisierul IN.TXT contine cel mult 1000000 numere naturale de cel mul 2 cifre fiecare. Sa se afiseze in fisierul OUT.TXT elementele din primul fisier, in ordine **crescatoare**.

**Ex.**

IN.TXT

79 45 67 4 3 22 45 79 45 4 3 4 3 3 4

OUT.TXT

3 3 3 3 4 4 4 4 22 45 45 67 79 79

**int v[100], x, mx;**

ifstream f (“IN.TXT”);

ofstream g (“OUT.TXT”);

int main()

{ while (f>>x) // se construieste vectorul de frecvente in timpul citirii numerelor din fisier

**v[x]++;**

// se afiseaza fiecare numar i de v[i] ori, parcurgand numerele i de maxim doua cifre

for (i=0; i<100; i++)

for (j=1; j<=v[i]; j++)

g<<i<<” ”;

return 0;

}

// se afiseaza fiecare numar i de v[i] ori, parcurgand numerele i de maxim doua cifre

for (i=0; i<100; i++)

while(v[i])

{ g<<i<<” ”;

v[i]--;

}

**Obs**. Deoarece construirea vectorului de frecvente se face in timpul citirii numerelor din fisier algoritmul este liniar, de complexitate O(n) -> eficient d.p.d.v al timpului de executare.

**Tema: 1005, 187, 244, 264, 267, 239, 247, 2669, 797, 2837, 809, 276, 2812**

**1005)**

ifstream fin("numere8.in");

ofstream fout("numere8.out");

int v[10000], x;

int main()

{

while(fin >> x)

if (x < 10000)

v[x] = 1;

for(int i = 9999; i > 0 ; i--)

if(v[i] == 0)

fout << i << ' ';

return 0;

}