

Structuri de control, divizori și lucru cu fișiere

// A.1) Afișarea primelor 20 numere naturale nenule cu WHILE

```
#include <iostream> // WHILE - afisare numere de la 1 la 20
```

```
using namespace std;
```

```
int main ( )
```

```
{ int i=1;
```

```
  while (i<=20) //cat timp i<=20 executa
```

```
  {
```

```
    cout <<i<<" ";
```

```
    i=i+1;
```

```
  }
```

```
  return 0;
```

```
}
```

// A.2) Afișarea primelor 20 numere naturale nenule cu DO-WHILE

```
#include <iostream> // DO-WHILE - afisare numere de la 1 la 20
```

```
using namespace std;
```

```
int main ( )
```

```
{ int i=1;
```

```
  do //cat timp i<=20 executa
```

```
  {
```

```
    cout <<i<<" ";
```

```
    i=i+1;
```

```
  } while (i<=20);
```

```
  return 0;
```

```
}
```

// A.3) Afișarea primelor 20 numere naturale nenule cu FOR

```
#include <iostream> // FOR - afisare numere de la 1 la 20
```

```
using namespace std;
```

```
int main()
```

```
{ int i;
```

```
  for (i=1;i<=20;i++) //Pentru i cu valori (din 1 in 1) de la 1 la 20
```

```
    cout <<i<<" "; //Afiseaza
```

```
  return 0;
```

```
}
```

// A.4) /* Se citeste un numar natural din FILE.IN

Sa se scrie in FILE.OUT cu spatiu intre ele toate numerele naturale nenule <=n */

```
#include <iostream>
```

```
#include <fstream> // Pentru a lucra cu fisiere
```

```
using namespace std;
```

```
ifstream f("FILE.IN"); // f - pentru citire din fisier
```

```
ofstream g("FILE.OUT"); //g - pentru scriere in fisier
```

```
int main()
```

```
{ int i,n;
```

```

f>>n; // citesc din f si salvez in n
f.close(); // inchid fisierul de intrare (nu mai am nevoie de date din el)

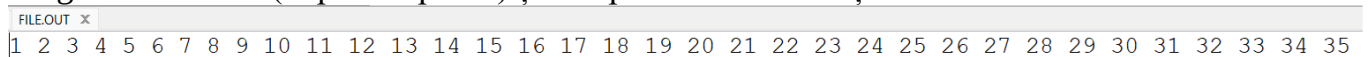
for (i=1;i<=n;i++)
    g<<i<<' ';// scriu in g valoarea curenta a lui i
g.close();
return 0;
}

```

Pentru următorul conținut al fișierului de intrare (care trebuie completat înainte de rularea programului):



Programul va crea (după compilare) și completa următorul fișier



```

//A.5)
/* Se citeste un numar natural din FILE.IN
   Sa se scrie toti divizorii acestui numar pe o linie in fisierul FILE.OUT separati cu un spatiu intre
   ei. Sa se scrie pe a 2-a linie a fisierului FILE.OUT numarul de divizori ai numarului citit din
   FILE.IN*/
#include <iostream>
#include <fstream> // pentru a lucra cu fisiere
using namespace std;
ifstream f("FILE.IN");
ofstream g("FILE.OUT");

int main()
{ int i,n,nrdiv=0; // n-nr citit din FILE.IN; i - posibil divizor; nrdiv - nr de divizori
  f>>n; // citesc din f si salvez in n
  f.close(); // inchid fisierul de intrare (nu mai am nevoie de date din el)

  for (i=1;i<=n;i++)
      if (n%i==0) //daca n la impartirea cu i da rest 0 atunci il scriu pe i in fisier
      {
          g<<i<<' ';// scriu in g valoarea lui i
          nrdiv++;// Nr divizorilor creste cu 1 (pentru ca am mai gasit un divizor)
      }
  g<<endl; // trec la randul urmator in FILE.OUT
  g<<nrdiv;
  g.close();
  return 0;
}

```

Dacă fișierul de intrare are următorul conținut:

```
FILE.IN x
35
```

Atunci, după rularea programului, fișierul de ieșire va conține:

```
FILE.OUT x
1 5 7 35
4
```

Aplicații:

1.) Se citește un număr natural din fișierul *numere.in*. Să se scrie în fișierul *numere.out* dacă numărul citit este prim.

Ex: dacă numere.in conține 7 atunci fișierul numere.out va conține mesajul 'este prim'.

2.) Se citește un număr natural **n** din fișierul *file.in* de pe prima linie. Să se scrie în *file.out* suma cifrelor lui **n**.

Ex: dacă file.in are conținutul:

745

atunci file.out va conține 16

3.) Se citesc 2 numere naturale **a** și **b** din fișierul *file.in* de pe prima linie. Să se scrie în *file.out* toate numerele pare dintre **a** și **b**. Dacă nu există numere pare între **a** și **b** atunci în fișierul *file.out* se va scrie -1.

Ex1: dacă file.in are conținutul:

7 18

atunci file.out va conține: 8 10 12 14 16 18

Ex2: dacă file.in are conținutul:

10 4

atunci file.out va conține: 4 6 8 10

Ex3: dacă file.in are conținutul:

7 7

atunci file.out va conține: -1