

3. Se citește un număr întreg nenul a ($a < 10^9$). Scrie un program C++ care afișează câte cifre pare are numărul a . În cazul în care numărul citit nu are cifre pare să se afișeze un mesaj corespunzător.

Exemple

Se citește: 45792041

Se afișează: 4

Se citește: 7139

Se afișează: **numarul nu are cifre pare**

4. Se citește un număr natural n ($n < 10^6$). Scrie un program C++ care afișează toți divizorii săi naturali în ordine crescătoare.

Exemplu

Se citește: 12

Se afișează: 1 2 3 4 6 12

5. Se citește un număr natural nenul n ($n < 10^6$). Scrie un program C++ care afișează cel mai mare număr natural de forma 2^k care este mai mic sau egal cu n .

Exemple

Se citește: 9

Se afișează: 8

Se citește: 1

Se afișează: 0

6. Se citește un număr natural n ($n < 10^6$). Scrie un program C++ care afișează un mesaj corespunzător dacă numărul este palindrom sau nu. Un număr se numește palindrom dacă răsturnatul său este egal cu el.

Exemple

Se citește: 19391

Se afișează: **Numarul este palindrom**

Se citește: 125251

Se afișează: **Numarul nu este palindrom**

7. Se citesc două numere naturale nenule a și b ($a < 10^6$ și $b < 10^6$). Scrie un program C++ care afișează cel mai mare divizor comun al lor.

Exemple

Se citesc: 9 12

Se afișează: 3

4.6.2. Instrucțiunea repetitivă **for**

Această instrucțiune se folosește atunci când știm de câte ori trebuie executat setul de instrucțiuni. Se mai numește instrucțiune repetitivă cu contor.

Instrucțiunea **for** este compusă din trei expresii ca în exemplele de mai jos:

a. **for (i=a; i<=b; i=i+c)** – variabila i are ca valoare inițială valoarea lui a și crește cu valoarea lui c până ajunge la valoarea lui b . Se folosește dacă $a <= b$.

b. `for (i=a; i>=b; i=i-c)` variabila `i` are ca valoare inițială valoarea lui `a` și scade cu valoarea lui `c` până ajunge la valoarea lui `b`. Se folosește atunci când `a>=b`.

În C++ mai există doi operatori pe care îi vei întâlni destul de des, mai ales atunci când programele conțin instrucțiuni `for`:

- `++` → incrementează valoarea variabilei cu `1`. Cu alte cuvinte, crește cu `1` valoarea variabilei.
- `--` → decrementează valoarea variabilei cu `1`. Cu alte cuvinte, scade cu `1` valoarea variabilei.

Printr-o analogie deducem că:

- `for (i=a; i<=b; i=i+1)` se mai poate scrie și `for (i=a; i<=b; i++)`
- `for (i=a; i>=b; i=i-1)` se mai poate scrie și `for (i=a; i>=b; i--)`

Exemple:

1. Se citește un număr natural `n` ($1 < n < 10^6$). Programul C++ care verifică dacă numărul este prim și afișează un mesaj corespunzător, ar putea fi:

```
#include <iostream>
using namespace std;
int main()
{
    int n, i, c=0;
    cout<<"Introduceți valoarea lui n: ";
    cin>>n;
    for(i=2; i<n; i++)
        if(n%i==0)
            c++;
    if(c==0)
        cout<<"Numarul este prim";
    else
        cout<<"Numarul nu este prim";
}
```

Modificați programul de mai sus, astfel încât să afișeze rezultatul așteptat pentru orice număr natural nenul $n < 10^6$;

Observație

Pentru o eficiență mai mare putem folosi `i*i<=n` în loc de `i<n`.

2. Se citește un număr natural nenul `a` ($a < 10^6$). Programul C++ care afișează toate numerele impare mai mici sau egale cu `a`, ordonate crescător, ar putea fi:

```
#include <iostream>
using namespace std;
int main()
{
    int n, i, c=0;
    cout<<"Introduceți valoarea lui n: ";
    cin>>n;
    for(i=1; i<=n; i=i+2)
        cout<<i<<" ";
}
```

Modificați programul de mai sus, astfel încât să afișeze toate numerele naturale pare mai mici sau egale cu `a`, în ordine descrescătoare.

Exerciții

1. Care dintre următoarele secvențe de instrucțiuni va afișa toate numerele naturale, nenule, mai mici sau egale cu n , în ordine crescătoare?

a. `for(i=1;i<=n;i--)`
`cout<<i<<" "`;

b. `for(i=0;i<=n;i++)`
`cout<<i<<" "`;

c. `for(i=n;i>=1;i--)`
`cout<<i<<" "`;

d. `for(i=1;i<=n;i++)`
`cout<<i<<" "`;

2. Scrie programul C++ corespunzător următorului algoritm, folosind instrucțiunea repetitivă `for`.

The Scratch script performs the following steps:

- When the green flag is clicked, ask for the value of 'a' and wait for a response.
- Set the variable 'a' to the response.
- Ask for the value of 'b' and wait for a response.
- Set the variable 'b' to the response.
- Set the variable 's' to 0.
- Set the variable 'nr' to 1.
- Set the variable 'i' to 'a'.
- Repeat the following loop until 'i' is greater than 'b':
 - Set 's' to 's + nr * i'.
 - Set 'nr' to '-1 * nr'.
 - Set 'i' to 'i + 1'.
- Speak the value of 's'.

3. Se citește un număr natural n ($n < 10^6$). Scrie un program C++ care verifică dacă numărul n este număr perfect și să se afișeze un mesaj corespunzător. Se numește număr perfect un număr care are suma divizorilor săi egală cu dublul său.

Exemple

Se citește: 6

Se afișează: **este numar perfect**

Se citește: 8

Se afișează: **nu este numar perfect**

4. Se citește un număr natural nenul n ($n < 10^6$). Scrie un program C++ care afișează toate numerele prime mai mici decât n , în ordine descrescătoare.

Exemplu

Se citește: 10

Se afișează: 7 5 3 2

5. Scrie un program C++ care citește un număr natural n ($n < 10^6$) și afișează toate pătratele perfecte mai mici sau egale cu n , în ordine descrescătoare.

Exemplu

Se citește: 17

Se afișează: 16 9 4 1

4.6.3. Instrucțiunea repetitivă `do ... while`

Instrucțiunea `do ... while` se mai numește și instrucțiune repetitivă condiționată posterior deoarece verificarea condiției se efectuează după executarea setului de instrucțiuni.

```
do
{
    instrucțiune1;
    instrucțiune2;
    .....
}while (condiție);
```

ATENȚIE!

Instrucțiunea repetitivă `do ... while` este singura după condiția căreia se pune `;`

Exemplu:

Se citește un număr natural n ($n < 10^6$). Programul C++ care afișează câte cifre are numărul n , poate fi:

```
#include <iostream>
using namespace std;
int main()
{
    int n, nr=0;
    cout<<"Introduceti valoarea lui n: ";
    cin>>n;
    do
    {
        nr++;
        n=n/10;
    }while (n!=0);
    cout<<"Numarul de cifre este: "<<nr;
}
```