

Cu număr cunoscut de pași 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.

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:

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`