

Algoritm de verificare a unui număr dacă este prim

Algoritmul de verificare a unui număr natural n dacă este prim:

- numerele 0, 1, respectiv numerele pare nu sunt prime
- vom verifica doar restul numerelor dacă sunt prime, adică pe 2 și numerele impare
- ne vom concentra asupra numerelor impare; pentru verificare vom folosi metoda reducerii la absurd – presupunem că numărul este prim și verificăm dacă presupunerea are loc. Numerele impare pot avea ca divizori doar numere impare, deci prima valoare pe care o vom verifica dacă este divizor este 3 și vom merge cu valorile pe care le verificăm dacă sunt divizori din 2 în 2.
- deci presupunem că numărul este prim
- pornim cu o variabilă d de la 3
- verificăm dacă d este divizor al lui n
- dacă d este divizor al lui n , atunci presupunerea făcută este falsă
- dacă d nu este divizor al lui n , îl creștem pe d cu 2 și verificăm din nou; pentru d mergem cel mult până la \sqrt{n} (dacă până aici nu am găsit divizori, nu vom găsi divizori nici pentru valori mai mari).

Algoritmul de verificare a unui număr n dacă este prim scris în pseudocod și C++.

Pseudocod

```
┌dacă  $n < 2$  sau  $n \% 2 = 0$  și  $n > 2$  atunci  $\text{prim} \leftarrow 0$ 
| altfel
|    $\text{prim} \leftarrow 1$ 
|    $d \leftarrow 3$ 
|   ┌cât timp  $d * d \leq n$  și  $\text{prim} = 1$  execută
|   |   ┌dacă  $n \% d = 0$  atunci  $\text{prim} \leftarrow 0$ 
|   |   | altfel  $d \leftarrow d + 2$ 
|   |   └─┬─┘
|   |   └─┬─┘
|   └─┬─┘
└─┬─┘
┌dacă  $\text{prim} = 1$  atunci scrie "numarul este prim"
| altfel scrie "numarul nu este prim"
└─┬─┘
```

C++

```
if(n < 2 || n % 2 == 0 && n > 2) prim = 0; //daca numarul este 1 sau 0 sau este par diferit de 2 nu e prim
else
{ prim = 1; //presupunem ca numarul este prim
//verificam daca n are divizori; pt ca este impar poate avea doar divizori impari, de aceea pornim de la 3
d = 3;
while(d * d <= n && prim == 1) //cat timp n-am depasit radical din numar si presupunerea este adevarata
{ if(n % d == 0) //daca d este divizor
prim = 0; //presupunerea ca n este prim este falsa
else d = d + 2; //daca d nu este divizor, se trece la urmatoarea valoare
}
if(prim == 1) //daca variabila prim a ramas cu valoarea 1, inseamna ca presupunerea facuta este adevarata
cout << "numarul este prim"; //se afiseaza mesajul ca numarul este prim
else cout << "numarul nu este prim"; //daca valoarea lui prim nu este 1 inseamna ca numarul nu este prim
```