Algoritmi cu numere prime

Pentru un număr n natural ≥ 2 să se stabilească dacă este prim.

Date de intrare – n (numărul citit)

Date de ieșire → „Număr prim".
→ „Nu este număr prim".

Analiza problemei.

Pornim analiza problemei de la definiția numărului prim.

Un număr este prim dacă se divide cu 1 și el însuși.

Vom verifica, prin numărare dacă numărul n are sau nu divizori proprii (divizori diferiți de 1 și de n). Pentru aceasta parcurgem intervalul numerelor naturale de la 2 la n/2, verificând pentru fiecare în parte dacă este sau nu divizor al lui n.

Pentru a obține rezultatul vom utiliza o variabilă sw cu semnificația – sw = 1, dacă n e prim și sw = 0, dacă n nu e prim. Pe parcursul verificării, dacă găsim un divizor, atunci variabilei sw îi atribuim valoarea 0. În momentul în care găsim un divizor, căutarea se poate încheia.

Algoritmul descris în limbajul pseudocod.

```
start
    citeste(n)
    sw ← 1
    pt. d=2, n/2, 1 execută
        dacă n%d=0 atunci
            sw=0
    dacă sw=1 atunci
        scrie("Numar prim")
        altfel
        scrie("Nu e numar prim".
stop.
```

O îmbunătățire a algoritmului se poate face în zona ciclului cu contor astfel:

```
    pt. d=2, n/2 și sw=1, 1 execută
        dacă n%d=0 atunci
            sw=0
```

O altă variantă de optimizare, găsiți în carte, la pagina 48.

2 aplicații cu numere prime

(A1) Să se afișeze toate numerele prime din intervalul $[2, a]$, unde a este un număr natural citit.

Exemplu: pt. a = 10 se afișează 2 3 5 7.

Date de intrare - a (limita superioară a intervalului)

Date de ieșire - n (numerele prime găsite).

Algoritmul în limbaj pseudocod.

```
start
  citește(a)
  pentru n=2, a, 1   // parcurg succesiv numerele de la 2 la a
    sw = 1
    // voi verifica fiecare nr. n dc. e prim
    pentru d=2, n/2, 1.
      dacă n%d=0 atunci
        sw=0
    dacă sw = 1 at.// nr. n e prim, având ca div. doar pe 1 și n
      scrie(n)
stop.
```

(A2) Să se numere câte numere prime există, mai mici decât un număr natural dat.

Date de intrare – x (numărul natural dat)

Date de ieșire – nrp (contorul care numără câte nr. prime s-au găsit

Algoritmul în limbaj pseudocod.

```
start
  citește(x)
  nrp = 0
  pentru i = 2, x, 1 // i ia fiecare nr. natural din intervalul [2, x]
      sw = 1
      // presupunem că i e nr. prim
      pentru j = 2, i/2, 1
          dacă i%j = 0 atunci
              sw = 0
      } verific dc. i e prim
      dacă sw = 1 atunci
          nrp = nrp + 1
  scrie(nrp)
stop.
```

Temă – pt. 31 martie

(1.) Alegeți una din cele 2 probleme prezentate și descrieți rezolvarea.

(2.) Problema rezolvată – Existență din manual, p. 52