

4. Se dă șirul de numere naturale: 7, 17, 37, 47, 67, 97, 107, 127, 137, 157 ... După identificarea metodei de generare a numerelor din șir, să se realizeze un pseudocod care afișează al n -lea termen al șirului dat, unde n este un număr natural nenul.

Date de intrare: n natural nenul;

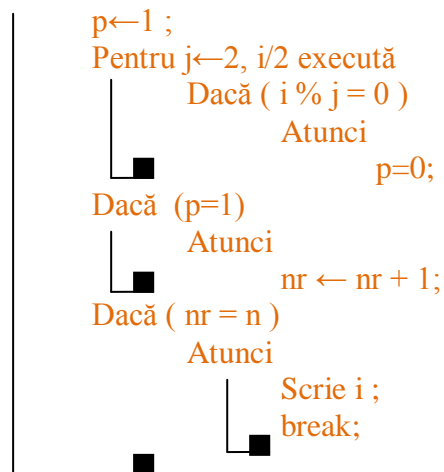
Date de manevră: nr, i, j, p;

Date de ieșire: al n -lea termen al șirului;

Citește n ;

nr ← 0;

Pentru $i \leftarrow 7, 2 * n, 10$ execută



5. Determinați dacă numărul natural nenul n este număr prim.

IN: n natural nenul;

OUT: da / nu;

D. manevră: p, i;

Observație: (p=0 NU) și (p=1 DA);

Citeste n ;

$p \leftarrow 1$; // presupunem că numărul este prim

Pentru $i \leftarrow 2, n/2$ execută

Dacă ($n \% i = 0$)

Atunci

$p = 0$;

break;

Dacă (p=1)

Atunci

Scrie „Da, n este număr prim.”;

Altfel

Scrie „Nu, n nu este număr prim.”;

**6. Un număr este perfect dacă el este egal cu suma divizorilor săi (exclusiv el însuși).
Scrieți un algoritm care să testeze dacă un număr natural dat este perfect.**

Date de intrare: n natural ;

Date de manevră: D, S (naturale); /* divizor ~ suma divizorilor */

Citește n;

$S \leftarrow 1$;

Pentru $D \leftarrow 2, n / 2$ execută

 Dacă ($n \% D = 0$) atunci

$S \leftarrow S + D$;

Dacă ($S = n$)

 Atunci

 Scrie "Numărul este perfect." ;

 Altfel

 Scrie "Numărul nu este perfect. " ;
