

HORSE - descrierea soluției

Se determină poziția numărului pe chenarul exterior al unei table reduse: știind că pe chenarul exterior al tablei de $n \times n$ sunt $8 \cdot (n-2)$ câmpuri, se diminuează p cu $8 \cdot (n-2)$ și n cu 4, până când $p \leq 8 \cdot (n-2)$ sau până când se ajunge în centru.

Pentru umplerea chenarului sunt necesare 4 tururi, iar pentru fiecare tur există 4 schimbări de direcție, deci se definesc 16 segmente de traseu, având – din păcate – lungimi și formule de calcul pentru linie și coloană, diferite.

De exemplu, pentru primul segment, $1 \leq p \leq [n/2]$, formulele de calcul sunt:

```
c:=2*p-1;
l:=2-p mod 2
```

<pre>const fi='horse.in';fo='horse.out'; var f:text; n,l,c,p,r:longint; procedure calc(var l,c:longint); begin if p<=(n-2)*2+2 then begin{primul tur} if p<=n div 2+1 then begin c:=p*2-1;l:=2-p mod 2;exit end; p:=p-n div 2; if p<=n div 2 then begin l:=p*2-1;c:=n-1+p mod 2;exit end; p:=p-n div 2; if p<=n div 2 then begin c:=n+2-p*2;l:=n-1+p mod 2;exit end; p:=p-n div 2; l:=n+2-p*2;c:=2-p mod 2;exit end; p:=p-(n-2)*2-2; if p<=(n-2)*2 then begin {al doilea tur} if p<=n div 2 then begin c:=p*2+1;l:=2-p mod 2;exit end; p:=p-n div 2; if p<=n div 2-1 then begin l:=p*2+2;c:=n-p mod 2;exit end; p:=p-n div 2+1; if p<=n div 2 -1 then begin c:=n-1-p*2;l:=n-1+p mod 2;exit end; p:=p-n div 2+1; l:=n-p*2;c:=2-p mod 2;exit end; p:=p-(n-2)*2; if p<2*(n-2) then begin {al treilea tur} if p<=n div 2 -1 then begin c:=p*2+2;l:=1+p mod 2;exit end; p:=p-n div 2+1; if p<= n div 2-1 then begin l:=p*2+2;c:=n-1+p mod 2;exit end; p:=p-n div 2+1; if p<=n div 2 then begin c:=n-p*2;l:=n-1+p mod 2;exit end; p:=p-n div 2; l:=n-1-p*2;c:=1+p mod 2;exit end; end;</pre>	<pre>{ultimul tur} p:=p-(n-2)*2+1; if p<=n div 2 -1 then begin c:=p*2+2;l:=2-p mod 2;exit end; p:=p-n div 2+1; if p<= n div 2 then begin l:=p*2+1;c:=n-1+p mod 2;exit end; p:=p-n div 2; if p<=n div 2-1 then begin c:=n-1-p*2;l:=n-p mod 2;exit end; p:=p-n div 2+1; l:=n-1-p*2;c:=2-p mod 2 end; begin assign(f,fi);reset(f); readln(f,n,p); close(f); r:=0; while (n>1) and(p>8*(n-2)) do begin p:=p-8*(n-2); dec(n,4); inc(r) end; l:=1;c:=1; if p>1 then calc(l,c); l:=l+2*r; c:=c+2*r; assign(f,fo);rewrite(f); writeln(f,l,' ',c); close(f) end.</pre>
--	--