Ciklinis skaičius

Tarkime, kad turime skaičių *n*. Patikrinti, ar jis ciklinis – nesunku. Skaičių išskaidome skaitmenimis ir užrašome į masyvą. Tuomet einame ratu kaip nurodyta sąlygoje, pasižymėdami visus skaitmenis, kuriuos aplankėme, kol kurį nors skaitmenį aplankome antrą kartą. Skaičius ciklinis, jei aplankytų skaitmenų kiekis lygus skaičiaus skaitmenų kiekiui ir jei tas antrą kartą aplankytasis skaitmuo yra pirmas skaičiaus skaitmuo.

Na o jei duotasis skaičius nėra ciklinis, teks perrinkti didesnius už jį skaičius, kol rasime tinkamą.

```
const ILGIS = 8;
type skaičius = array [1..ILGIS] of 0..9;
     lentelė = array [1..ILGIS] of boolean;
                        { skaitmenų, įeinančių į seką lentelė }

function ciklinis (sk: skaičius; kiek: integer): boolean;
{ ar skaičius, turintis „kiek“ skaitmenų yra ciklinis }
  var būta: lentelė;
      nariai, { kiek skaitmenų įtraukta į seką }
      i: integer;
begin
  for i := 1 to kiek do
    būta[i] := false;   { seka kol kas tuščia }
  nariai := 0;          { ji neturi nei vieno nario }
  i := 1;
  while not būta [i] do
    begin
     būta[i] := true;
     nariai := nariai + 1;
     i := (i+sk[i]-1) mod kiek + 1;{ pereiname per sk[i] skaitmenų }
    end;
  ciklinis := (nariai = kiek) and (i = 1)
end; { ciklinis }
```

```
procedure užrašyti (n: longint;
                    var sk: skaičius; var kiek: integer);
{ skaičių n užrašo masyvu ir randa jo skaitmenų kiekį }
  var i, nn: longint;
begin
  { randame n skaitmenų skaičių }
  kiek := 0;
  nn := n;
  while nn > 0 do
    begin
      kiek := kiek + 1;
      nn := nn div 10
     end;
  { skaičių užrašome masyvu }
  for i := kiek downto 1 do
    begin
      sk[i] := n mod 10;
      n := n div 10
    end
end; { užrašyti }

function rasti (n: longint): longint;
{ randa mažiausią ciklinį skaičių, ne mažesnį už n }
  var sk: skaičius;
      kiek: integer; { kiek skaičius turi skaitmenų }
begin
  n := n - 1;
  repeat
    n := n + 1;
    užrašyti (n, sk, kiek);
  until ciklinis (sk, kiek);
  rasti := n;
end; { rasti }
```