Plokštelės su skylutėmis

Norint išspręsti šį uždavinį reikia mokėti dvimačius masyvus. Uždavinys nereikalauja kokio nors įmantresnio algoritmo. Tereikia kruopščiai atlikti patikrinimą.

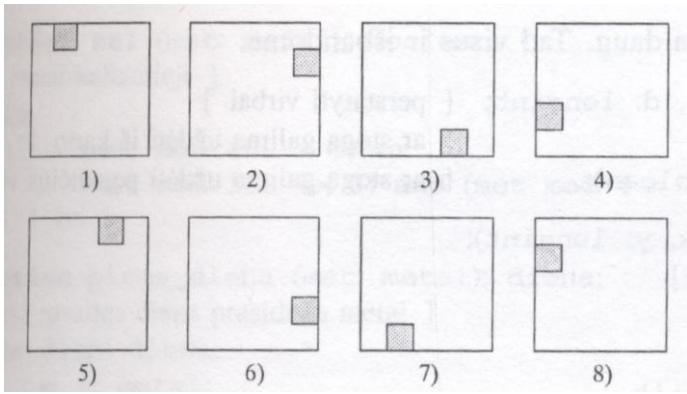
Reikia patikrinti aštuonis variantus: keturi variantai gaunami vis pasukant antrąją plokštelę 90 laipsnių kampu, ir keturi variantai – apvertus antrąją plokštelę.

Tikrinti galima dviem būdais:

1) sukti (versti) antrąją plokštelę ir tada ją dėti ant pirmosios;

2) iš karto skaičiuoti, kiek kiekvienu iš aštuonių atvejų liks neuždengtų skylučių. Tam reikia peržiūrinėti visus pirmosios plokštelės kvadratėlius. Radus skylutę patikrinti, kuriais iš aštuonių variantų skylutė bus uždengta. Jei pirmosios plokštelės skylutės koordinatės yra (i, j), tai galimus aštuonis variantus atitiks šie antrosios plokštelės kvadratėliai:

1) nepasukta plokštelė: (i, j);
2) pasukta 90° kampu pagal laikrodžio rodyklę $(j, n - i + 1)$;
3) pasukta 180° kampu pagal laikrodžio rodyklę $(n - i + 1, n - j + 1)$;
4) pasukta 270° kampu pagal laikrodžio rodyklę $(n - j + 1, i)$;
5) apversta ir nepasukta plokštelė: $(i, n - j + 1)$;
6) apversta ir pasukta 90° kampu pagal laikrodžio rodyklę $(n - j + 1, n - i + 1)$;
7) apversta ir pasukta 180° kampu pagal laikrodžio rodyklę $(n - i + 1, j)$;
8) apversta ir pasukta 270° kampu pagal laikrodžio rodyklę (j, i);



```
const MAX = 100;  { maksimalus plokštelės dydis }
type plokštelė = array [1..MAX, 1..MAX] of boolean;
                 { true – kvadratėlis iškirptas }

function skylutės (n: integer; var pl1, pl2: plokštelė)
                                                  : integer;
  var s: array [1..8] of integer;
      { kiek skylučių matosi uždėjus antrąją plokštelę ant pirmosios }
      { atitinkamai apvertus ar pasukus antrąją plokštelę }
      i, j, min: integer;
begin
  for i := 1 to 8 do
    s[i] := 0;
  for i := 1 to n do
    for j := 1 to n do
       if pl1[i, j]  { jei skylutė pirmajame kvadratėlyje }
       then begin
             if pl2[i, j] then s[1] := s[1] + 1;
             if pl2[j, n-i+1] then s[2] := s[2] + 1;
             if pl2[n-i+1, n-j+1] then s[3] := s[3] + 1;
             if pl2[n-j+1, i] then s[4] := s[4] + 1;
             if pl2[i, n-j+1] then s[5] := s[5] + 1;
             if pl2[n-j+1, n-i+1] then s[6] := s[6] + 1;
             if pl2[n-i+1, j] then s[7] := s[7] + 1;
             if pl2[j, i] then s[8] := s[8] + 1;
            end;
  min := s[1];
  for i := 2 to 8 do
    if s[i] < min
       then min := s[i];
  skylutės := min;
end;  { skylutės }
```