

sablon

Problema se rezolvă relativ simplu ținând cont de următoarele observații:

1. Deoarece întregul mesaj a fost codificat prin 4 rotiri ale textului, este clar că la o poziționare a textului sub șablon pot fi citite $Lung(Mesaj)/4$ caractere, deci întregul mesaj are $4 * NumarGăuri$ caractere
2. Ca urmare a observației de la punctul 1, mesajul poate fi împărțit exact în 4 șiruri de lungimi egale $Mesaj_1, \dots, Mesaj_4$
3. Dacă o gaură se află în poziția $T[i, j]$ din șablon, ei îi corespund pozițiile
 $T[j, N-i+1]$ la rotire cu 90 grade
 $T[N-i+1, N-j+1]$ la rotire cu 180 grade
 $T[N-j+1, i]$ la rotire cu 270 grade
de unde deducem că nu e nevoie să rotim textul!!!
4. Dacă lungimea unui șir este L_4 (vezi în sursă), este suficient să parcurgem numai primul din cele 4 șiruri cu un Index. Atunci, parcurgând textul care ascunde mesajul, în poziția (i, j) există o gaură în șablon dacă și numai dacă toate cele 4 caractere
 $Mesaj_1[Index], Mesaj_2[Index], Mesaj_3[Index], Mesaj_4[Index]$
coincid cu cele 4 caractere obținute prin rotire (vezi observația 3)
5. "Cel mai bun pseudocod este... PASCAL-ul", deci:

```
Program DeterminareSablon;
Const DimMax = 50;           { lungimea maxima a textului }
Type Gauri = Array [1..DimMax, 1..DimMax] Of Boolean;
    Textul = Array [1..DimMax, 1..DimMax] Of Char;

Var G : Gauri;
    T : Textul;
    Mesaj: Array [1..4] Of String;
    N, L4: Integer;

Procedure Afisare;
Var i, j: Integer;
    f : Text;
Begin
    Assign(f, 'sablon.out'); Rewrite(f);
    For i := 1 To N Do
        Begin
            For j := 1 To N Do
                If G[i,j] Then Write(f, 'O')
                Else Write(f, 'X');
            Writeln(f);
        End;
    Close(f);
End;

Procedure Prelucrare;
Var Index, i, j, Lung, N2: Integer;
Begin
    FillChar(G, SizeOf(G), False);
    Index := 1;           {parcurs cu Index pana la length(mesaj)/4-pentru 4 rotiri}
    For i := 1 To N Do
        For j := 1 To N Do
            Begin
                If (Mesaj[1][Index] = T[i,j]) And {primul caracter}
                    (Mesaj[2][Index] = T[j,N-i+1]) And {rotit cu 90°}
                    (Mesaj[3][Index] = T[N-i+1,N-j+1]) And {rotit cu 180°}
                    (Mesaj[4][Index] = T[N-j+1,i]) {rotit cu 270°}
                Then {daca toate 4 caractere coincid}
                    Begin
                        G[i,j] := True; {e gaura}
                        Inc(Index); {trec la alt caracter}
                        If Index > L4 Then exit; {div 4 pentru 4 rotiri}
                    End;
            End;
        End;
    End;
End;
```

```

Procedure Citire;
Var f    : Text;
      i, j: Integer;
      Lung: Integer;
      M    : Array[1..1000] Of Char;
Begin
  Assign(f, 'sablon.in'); reset (f);
  Lung := 0;
  FillChar(M, SizeOf(M), #0);
  While Not Eoln(f) Do           {citesc mesajul in sirul de caractere}
    Begin                         {determinandu-i lungimea}
      Inc(Lung);
      Read(f, M[Lung]);
    End;
  ReadLn(f, N);                   {citesc N}
  For i := 1 To N Do             {citesc textul in T}
    Begin
      For j := 1 To N Do
        Read(f, T[i,j]);
      ReadLn(f);
    End;
  Close(f);
  If Lung Mod 4 <> 0 Then
    WriteLn('Mesajul nu are lungimea corecta')
  Else
    Begin
      L4 := Lung Div 4;           {determin numarul de caractere la o rotire}
      For i := 1 To L4 Do         {si impart mesajul in 4 parti egale}
        Begin
          Mesaj[1][i] := M[i];
          Mesaj[2][i] := M[L4+i];
          Mesaj[3][i] := M[2*L4+i];
          Mesaj[4][i] := M[3*L4+i]
        End
      End
    End;

Begin
  Citire;
  Prelucrare;
  Afisare;
End.

```