

# Problema 1 – descriere soluție

## CEZAR

Prof. Șerban Marinel

Se reține alfabetul într-un vector constant cu 26 de elemente.

Se rețin pozițiile spațiilor într-un vector cu maxim 127 – am presupus că cel mai 'ciudat' mesaj poate fi de forma L L L L ... L (adică Literă spațiu Literă spațiu...).

Se rețin, de asemenea, într-un vector cele 10 chei.

Se parcurge apoi mesajul, reținând pe rând pozițiile spațiilor și eliminându-le. După eliminarea acestora se trece la prelucrarea mesajului, pe rând după cele 10 chei.

Se ține cont că prima coloană este formată din caracterele cu numerele de ordine 1, 11, 21, ... din mesaj, coloana a doua din cele cu numerele de ordine 2, 12, 22, ..., etc.

Acest lucru înseamnă faptul că nu e nevoie să utilizăm efectiv un tablou bidimensional, ci doar să formăm pe rând coloanele.

Apoi coloana formată se decodifică ținând cont de faptul că pentru litera aflată în alfabet pe poziția Pozitie, caracterul cu care se va înlocui se găsește în alfabet pe poziția

$Pozitie\_noua := (Pozitie - k + 26) \text{ Mod } 26$ , unde k este cheia de codificare.

Se reface mesajul și se pun spațiile înapoi la locul lor.

**Program** Cezar;

```
CONST Alfabet: Array[0..25] Of Char=('A','B','C','D','E','F','G','H','I','J',
                                         'K','L','M','N','O','P','Q','R','S','T',
                                         'U','V','W','X','Y','Z');

Var Mesaj : String;                                {mesajul codificat}
    Spatii: Array[1..127] Of Integer;                {pot fi maxim 127 spatii pe pozitii<255}
    K      : Array[1..10] Of Integer;                {cele 10 chei de codificare}
    i, L   : Integer;
    Fin, Fout: Text;
    NrSpatii : Integer;

Procedure Retine_spatii;
Var i: Integer;
Begin
    NrSpatii := 0;
    i := Pos(' ', Mesaj);                            {caut primul spatiu}
    While i>0 Do                                     {cat timp exista}
        Begin
            Mesaj[i] := '_';                            {il inlocuiesc cu '_'}
            NrSpatii := NrSpatii+1;                      {retin pozitia spatiului}
            Spatii[NrSpatii] := i;
            i := Pos(' ', Mesaj)                        {caut urmatorul spatiu}
        End;
    i := Pos('_', Mesaj);                              {caut primul '_'}
    While i>0 Do                                     {cat timp exista}
        Begin
            Mesaj := Copy(Mesaj, 1, i-1)+Copy(Mesaj, i+1, 255); {il elimin}
            i := Pos('_', Mesaj)                        {caut urmatorul '_'}
        End;
    End;

Procedure Codifica_mesaj;
Var Coloana: String;
    i: Integer;

    Procedure Fa_coloana(i: Integer);
    Var j: Integer;
    Begin                                                {formez coloana cu elemente din 10 in 10}
        Coloana := '';
        j := i;                                          {primul fiind elementul de pe pozitia i}
        While j<=L Do
            Begin
                Coloana := Coloana+Mesaj[j];
                j := j+10
            End;
    End;
```

```

Procedure Codifica(Var C: String; k: Integer);
Var i, Pozitie, Pozitie_noua: Integer;
Begin
    For i := 1 To Length(C) Do
        Begin
            Pozitie := Ord(C[i])-65;           {pozitia actuala a caracterului in alfabet}
            Pozitie_noua := (Pozitie-k+26) Mod 26; {noua pozitie in alfabet}
            C[i] := Alfabet[Pozitie_noua]      {se inlocuieste caracterul cu cel nou}
        End;
    End;

Procedure Desfa_coloana(i: Integer);
Var j, n: Integer;
Begin
    j := i; n := 1;
    While n<=Length(Coloana) Do           {refac mesajul punand elementele la locul lor}
        Begin
            Mesaj[j] := Coloana[n];
            n := n+1;
            j := j+10
        End;
    End;

Begin
    For i := 1 To 10 Do                   {pentru fiecare din cele 10 chei}
        Begin
            Fa_coloana(i);                  {construieste coloana corespunzatoare cheii i}
            Codifica(Coloana, K[i]);        {decodifica coloana utilizand cheia i}
            Desfa_coloana(i);               {repune elementele la locul lor}
        End;
    End;

Procedure Pune_spatii;
Var i: Integer;
Begin
    For i := 1 To NrSpatii Do             {pune spatiile inapoi la locul lor}
        Mesaj := Copy(Mesaj, 1, Spatii[i]-1)+' '+Copy(Mesaj, Spatii[i], 255)
    End;

Begin
    Assign(Fin, 'cezar.in'); Reset(Fin);
    Assign(Fout, 'cezar.out'); Rewrite(Fout);
    ReadLn(Fin, Mesaj);
    L := Length(Mesaj);
    For i := 1 To 10 Do
        Read(Fin, K[i]);
    Close(Fin);
    Retine_spatii;
    Codifica_mesaj;
    Pune_spatii;
    WriteLn(Fout, Mesaj);
    Close(Fout);
End.

```