

4. uzdevums [uzd4] (10 punkti)

Sastādi programmu, kas realizē doto algoritmu. Ņem vērā, ka masīva pirmā elementa kārtas indekss ir 1.

- 1. solis.** Lietotājs ievada veselu skaitli n , intervālā no 10 līdz 20.
- 2. solis.** Masīva $A[n]$ vērtības aizpilda ar nejaušiem skaitļiem intervālā no 10 līdz 50.
- 3. solis.** Ekrānā izvada masīva $A[n]$ elementus.
- 4. solis.** $A[1]$ vērtību saglabā mainīgajā $m1$.
- 5. solis.** $A[2]$ vērtību saglabā mainīgajā $m2$.
- 6. solis.** Mainīgajam k piešķir vērtību 3.
- 7. solis.** Ja masīva $A[k]$ elementa vērtība ir mazāka nekā $m1$, tad $m1$ piešķir $A[k]$ elementa vērtību.
- 8. solis.** k vērtību palielina par 1.
- 9. solis.** Ja masīva $A[k]$ elementa vērtība ir lielāka nekā $m2$, tad $m2$ piešķir $A[k]$ elementa vērtību.
- 10. solis.** k vērtību palielina par 1.
- 11. solis.** Ja k vērtība ir mazāka par $n+1$, pāriet pie 7.soļa izpildes, citādi pāriet pie 12.soļa izpildes.
- 12. solis.** Ekrānā izvada $m1$ vērtību.
- 13. solis.** Ekrānā izvada $m2$ vērtību.
- 14. solis.** Mainīgajam m piešķir mainīgo $m1$ un $m2$ summu.
- 15. solis.** Ekrānā izvada n vērtību.

Ko dara (ko spēj aprēķināt) dotais algoritms?

5. uzdevums [uzd5] (10 punkti)

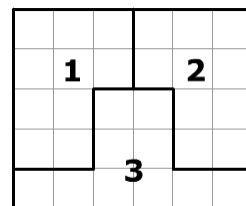
Uz planētas, kuru apdzīvo roboti, iedzīvotāji nemīl lietot decimālo skaitīšanas sistēmu, tādēļ, lai sazinātos ar viņiem, visi decimālie cipari ir jāaizstāj ar binārajiem cipariem. Piemēram, ja lietotāja ievadītais ziņojuma teksts ir, **HG2kj7##1NZ4**, tad pārveidotais ziņojums ir **HG10kj111##1NZ100**.

Sastādīt programmu, kas lietotāja ievadīto ziņojumu – simbolu virkni, kura garums nepārsniedz 250 simbolus, pārveido un izdrukā ekrāna robotiem pieņemamā veidā.

6. uzdevums [uzd6] (10 punkti)

Kārlis topošo arhitektu konkursam veido maketu. Tā kā materiāls, kuru viņš plāno izmantot maketa sienu izgatavošanai nav lēts, viņam iepriekš jāveic plānošanas un aprēķina darbi.

Kārlis uz rūtiņu papīra uzzīmējis ēkas starpsienu plānojumu. Zināms, ka vienas rūtiņas izmērs dzīvē būs 10 cm x 10 cm, bet sienu augstums būs 30 cm.



Sastādi programmu, kas palīdz Kārlim aprēķināt (izdrukā failā *uzd6.out*), cik kvadrātcentimetrus materiāla nepieciešams iegādāties, lai izgatavotu iecerēto maketu, ja faila *uzd6.in* pirmajā rindā doti divi veseli skaitļi, kas atdalīti ar vienu atstarpi, n un m ($1 \leq n, m \leq 100$) - ēkas izmērs rūtiņās, bet katrā no sekojošajām faila n rindiņām doti m skaitļi atdalīti ar atstarpi, kas apzīmē katras rūtiņas piederību telpai plānā.

Piemēram,

ievaddati *uzd6.in*

6 5

1 1 1 2 2 2

1 1 1 2 2 2

1 1 3 3 2 2

1 1 3 3 2 2

3 3 3 3 3 3

izvaddati *uzd6.out*

10200

7. uzdevums [uzd7] (15 punkti)

Izmantojot četrus ciparus iespējams pierakstīt pulksteņa laiku formātā HH:MM. Sastādīt programmu, kas izdrukā ekrānā visus iespējamus pulksteņa laikus, ko var izveidot no lietotāja ievadītajiem četriem cipariem, katru no tiem izmantojot tikai vienu reizi.

Piemēram,

- ja lietotājs ievada 1, 2, 3 un 4, programma izdrukā
12:34 12:43 13:24 13:42 14:23 14:32 21:34 21:43 23:14 23:41
- ja lietotājs ievada 0, 1, 8 un 8, programma izdrukā
08:01 08:10 18:08

8. uzdevums [uzd8] (15 punkti)

Bērni nolēma spēlēt paslēpes. Tā kā neviens nevēlējās būt pirmais, kurš pārējos meklēs, bērni nolēma nostāties aplī un skaitīt skaitāmpantiņu. Katrs bērns (pulksteņrādītāja virzienā) secīgi skaitīja pa vienam skaitāmpantiņa vārdam. Tam, kurš izrunāja skaitāmpantiņa pēdējo vārdu, bija pirmajam jāmeklē citus.

Sastādīt programmu, kas nosaka, kuram no bērniem pirmajam jāmeklē citus, ja faila *uzd8.in* pirmajā rindā dots skaitāmpantiņš, bet otrajā rindā – ar vienu atstarpi atdalīti bērnu vārdi, ne vairāk kā 50. Bērnu vārdu, kuram jāmeklē pirmajam, ierakstīt failā *uzd8.out*.

Piemēram:

ievaddati *uzd8.in*

Ennik, bennik, sikel, sa, sesi viri pagraba. Viens, divi, trīs, nu tu esi brivs.

Markuss Laine Marta Juris Santa Gatis Reinis Linda

izvaddati *uzd8.out*

Gatis

9. uzdevums [uzd9] (10 punkti)

Biatlona sacensību finālā ar atsevišķo startu tiesības piedalīties bija izcīnījuši 12 sportisti. Pēc tam, kad katrs no dalībniekiem finišēja (sportisti finišēja secībā, kas atbilda viņu kārtas numuriem – skaitlis no 1 līdz 12), skatītājiem tika paziņots, kādu vietu dotajā brīdī ieņem finišējušais sportists.

Sastādīt programmu, kas nolasa failu *uzd9.in*, kurā doti 12 skaitļi vienā rindā, atdalīti ar vienu atstarpi, kas norāda sportistu izcīnīto vietu finišēšanas brīdī. Noteikt sacensību rezultātus. Faila *uzd9.out* pirmajā rindā izvadīt pirmā sportista kārtas numuru un pēc vienas atstarpes sacensībās izcīnīto vietu, otrajā rindā otrā sportista kārtas numuru un pēc vienas atstarpes sacensībās izcīnīto vietu, ..., divpadsmitajā rindā divpadsmitā sportista kārtas numuru un pēc vienas atstarpes sacensībās izcīnīto vietu.

Piemēram,

ievaddati *uzd9.in*

1 1 2 3 1 4 4 2 1 3 1 2

izvaddati *uzd9.out*

1 12

2 7

3 8

4 11

5 4

6 10

7 9

8 6

9 3

10 5

11 1

12 2

10. uzdevums [uzd10] (10 punkti)

Ralfs skolas kursā "Programmēšanas pamati" šobrīd apgūst tēmu "Saspiešanas algoritmi". Skolotājs viņam ir iedevis programmas piemēru, kas realizē datu saspiešanu pēc RLE algoritma, tas ir: saspiežot informāciju, simbolu, kas atkārtojas vairākas reizes, raksta vienu reizi un aiz simbola raksta skaitli, kas norāda, cik reizes simbols atkārtojas.

Piemēram,

ja saspiežamā simbolu virkne ir:

aaaaabbbaccbbddddd,

tad saspiešanās simbolu virknes pieraksts ir:

a5b3a1c3b1d5.

Palīdzi Ralfam uzrakstīt programmu, kas no saspiešanās simbolu virknes, kas dota teksta failā *uzd10.in*, atjauno oriģinālo simbolu virkni un ieraksta to failā *uzd10.out*.

RLE algoritma realizācija

(Pascal kods)

```
program RLE;
var dati1,dati2:text;
    s1,s2,r:char;
    sk:integer;
begin
assign(dati1,'uzd10.in');
reset(dati1);
assign(dati2,'uzd10.out');
rewrite(dati2);
read(dati1,s1);
if EOF(dati1) then write(dati2,s1,1);
while (not EOF(dati1)) do begin
    read(dati1,s2);
    sk:=1;
    while (s1=s2)and(not EOF(dati1)) do
        begin
            sk:=sk+1;
            read(dati1,s2);
        end;
    if (EOF(dati1)) and (s1=s2) then sk:=sk+1;
    write(dati2,s1,sk);
    if (EOF(dati1)) and (s1<>s2) then write(dati2,s2,1);
    s1:=s2;
end;
close(dati1);
close(dati2);
end.
```

RLE algoritma realizācija

(C++ kods)

```
#include <fstream>
using namespace std;
int main(){
    ifstream dati1;
    ofstream dati2;
    dati1.open("uzd10.in",ios::in);
    dati2.open("uzd10.out",ios::out);
    char s1,s2;
    int sk;
    dati1.get(s1);
    while(!dati1.eof()){
        sk=1;
        dati1.get(s2);
        while(s1==s2&&!dati1.eof()){
            sk++;
            dati1.get(s2);
        }
        dati2<<s1<<sk;
        s1=s2;
    }
    dati1.close();
    dati2.close();
}
```

**RLE algoritma realizācija
(VisualBasic kods)**

```
Sub Main()  
    Dim dati1 As String  
    Dim dati2 As System.IO.StreamWriter  
    dati1=My.Computer.FileSystem.  
        ReadAllText("uzd10.in")  
    dati2=My.Computer.FileSystem.  
        OpenTextFileWriter("uzd10.out",False)  
    If Len(dati1) > 0 Then  
        Dim s1, s2 As Char  
        Dim nr, sk As Integer  
        nr = 0  
        s1 = dati1(nr)  
        nr = nr + 1  
        While nr < Len(dati1)  
            sk = 1  
            s2 = dati1(nr)  
            nr = nr + 1  
            While (s1 = s2 And nr < Len(dati1))  
                sk = sk + 1  
                s2 = dati1(nr)  
                nr = nr + 1  
            End While  
            If (nr = Len(dati1)) And  
                (s1 = s2) Then sk = sk + 1  
            dati2.Write(s1 & sk)  
            If (nr = Len(dati1)) And  
                (s1 <> s2) Then dati2.Write(s2 & 1)  
            s1 = s2  
        End While  
        If (nr = Len(dati1)) And (s1 <> s2)  
            Then dati2.Write(s1 & 1)  
        End If  
        dati2.Close()  
    End Sub
```

**RLE algoritma realizācija
(Java kods)**

```
import java.io.BufferedReader;  
import java.io.BufferedWriter;  
import java.io.FileReader;  
import java.io.FileWriter;  
import java.io.IOException;  
public class RLE algoritms {  
    public static void main(String[] args) {  
        try {  
            char s1,s2;  
            int sk;  
            int x;  
            BufferedReader dati1 = new  
                BufferedReader(new FileReader("uzd10.in"));  
            BufferedWriter dati2 = new  
                BufferedWriter(new FileWriter("uzd10.out"));  
            x=dati1.read();  
            s1 = (char) x;  
            while (x != -1){  
                sk = 1;  
                x = dati1.read();  
                s2 = (char) x;  
                while(s1==s2 && x != -1){  
                    sk++;  
                    x = dati1.read();  
                    s2 = (char) x;  
                }  
                dati2.write(s1+Integer.toString(sk));  
                s1=s2;  
            }  
            dati1.close();  
            dati2.close();  
        } catch (IOException e) {}  
    }  
}
```

**RLE algoritma realizācija
(VBA kods)**

```
Sub RLE()  
  Open "uzd10.in" For Input As #1  
  Dim dati1 As String  
  Input #1, dati1  
  Close #1  
  If Len(dati1) > 0 Then  
    Dim s1, s2 As String  
    Dim nr, sk As Integer  
    Dim dati2 As String  
    nr = 1  
    s1 = Mid(dati1, nr, 1)  
    nr = nr + 1  
    Do While nr <= Len(dati1)  
      sk = 1  
      s2 = Mid(dati1, nr, 1)  
      nr = nr + 1  
      Do While (s1 = s2) And (nr <= Len(dati1))  
        sk = sk + 1  
        s2 = Mid(dati1, nr, 1)  
        nr = nr + 1  
      Loop  
      If (nr > Len(dati1)) And (s1 = s2) Then sk = sk + 1  
      dati2 = dati2 + s1 + CStr(sk)  
      If (nr = Len(dati1)) And (s1 <> s2) Then dati2 = dati2 + s2 + "1"  
      s1 = s2  
    Loop  
    If (nr = Len(dati1)) And (s1 <> s2) Then dati2 = dati2 + s1 + "1"  
  End If  
  Open "udz10.out" For Output As #1  
  Print #1, dati2  
  Close #1  
End Sub
```