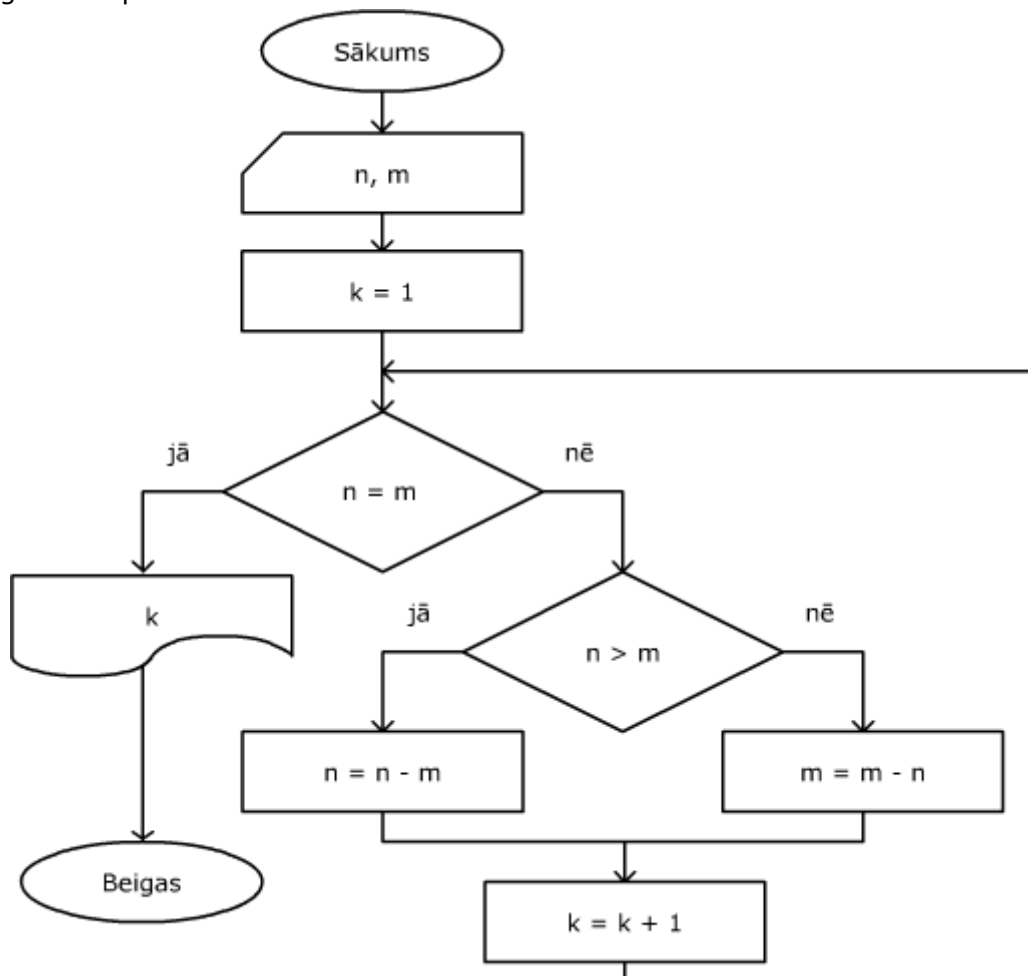


Liepājas Universitātes  
**Datorzinātņu olimpiādes**  
skolēnu programmēšanas konkursa uzdevumi

**1. uzdevums [uzd1]** (5 punkti)

Sastādi programmu pēc dotās blokhēmas.



**2. uzdevums [uzd2]** (5 punkti)

Uzzīmē blokhēmu vai apraksti algoritmu strukturētā teksta veidā, pēc kura būtu iespējams noteikt, cik riņķa līnijas iespējams uzzīmēt uz taisnstūra veida lapas, lai tās nekrustojas, ja  $n$  un  $m$  ir taisnstūra lapas malu garumi, bet  $r$  riņķa līnijas rādiuss.

**3. uzdevums [uzd3]** (10 punkti)

Aleksai ir kvadrātveida papīra lapa, kuras malu garumi ir  $n$ . Sākumā Aleksa lapu sagrieza četrās vienādās daļās, iegūstot četrus vienādus trijstūrus. Pēc tam, viņa katru trijstūri pārgrieza uz pusēm, iegūstot no katra trijstūra divus vienādus trijstūrus. Hm ..., cik trijstūru iegūtu, ja vēl  $n$  reizes katru trijstūri pārgrieztu uz pusēm?

Sastādi programmu, kas aprēķina un izvada ekrānā Aleksas iegūto trijstūru skaitu, ja lietotājs ievada kvadrāta malas garumu  $n$  ( $1 \leq n \leq 20$ ).

**4. uzdevums [uzd4]** (15 punkti)

Maizniekam ir ticējums, lai kūka sanāktu perfekti simetriska, tā no cepeškrāsns jāizņem brīdī, kad pulkstenis rāda palindroma laiku, tas ir, pulksteņa rādījums, lasot no kreisās uz labo pusi, ir vienāds ar pulksteņa rādījumu, kas iegūts, lasot to no labās uz kreiso pusi.

Sastādi programmu, kas nosaka un izvada ekrānā tuvāko palindroma laiku, ja zināms pulksteņa rādījums (ievada lietotājs formā HH:MM) brīdī, kad kūka tiek ielikta cepeškrāsnī. Piemēram,

- ja lietotājs ievada 00:00, programma izvada 01:10;
- ja lietotājs ievada 12:34, programma izvada 13:31;
- ja lietotājs ievada 14:15, programma izvada 14:41;
- ja lietotājs ievada 23:57, programma izvada 01:10.

### 5. uzdevums [uzd5] (10 punkti)

Sastādi programmu, kas realizē doto algoritmu. Ņem vērā, ka masīva A un B pirmā elementa kārtas indekss ir viens, bet C - nulle.

- 1.solis.** Lietotājs ievada divus veselus skaitļus –  $n$  un  $m$ , intervālā no 1 līdz 10.
- 2.solis.** Masīva  $A[n]$  vērtības aizpilda ar nejaušiem skaitļiem intervālā no 0 līdz 1.
- 3.solis.** Masīva  $B[m]$  vērtības aizpilda ar nejaušiem skaitļiem intervālā no 0 līdz 1.
- 4.solis.** Masīva  $C[k]$  vērtības aizpilda ar 0.
- 5.solis.** Masīva  $C[k]$  elementam piešķir masīva  $A[n]$ ,  $B[m]$  un  $C[k]$  elementa summu.
- 6.solis.** Ja masīva  $C[k]$  elementa vērtība ir lielāka nekā 1, masīva  $C[k]$  elementa vērtību samazina par 2 un masīva  $C[k-1]$  elementa vērtību aizstāj ar 1.
- 7.solis.**  $n$  vērtību samazina par 1.
- 8.solis.**  $m$  vērtību samazina par 1.
- 9.solis.**  $k$  vērtību samazina par 1.
- 10.solis.** Ja  $k$  vērtība ir lielāka nekā 0, pāriet pie 5.soļa izpildes, citādi pāriet pie 11.soļa izpildes.
- 11.solis.** Ekrānā izvada masīva  $A[n]$  elementus.
- 12.solis.** Ekrānā izvada pluss zīmi.
- 13.solis.** Ekrānā izvada masīva  $B[m]$  elementus.
- 14.solis.** Ekrānā izvada vienādības zīmi.
- 15.solis.** Ekrānā izvada masīva  $C[k]$  elementus.

Ko dara (ko spēj aprēķināt) dotais algoritms?

Kāda ir jābūt  $k$  sākuma vērtībai attiecībā pret  $n$  un  $m$  sākuma vērtību?

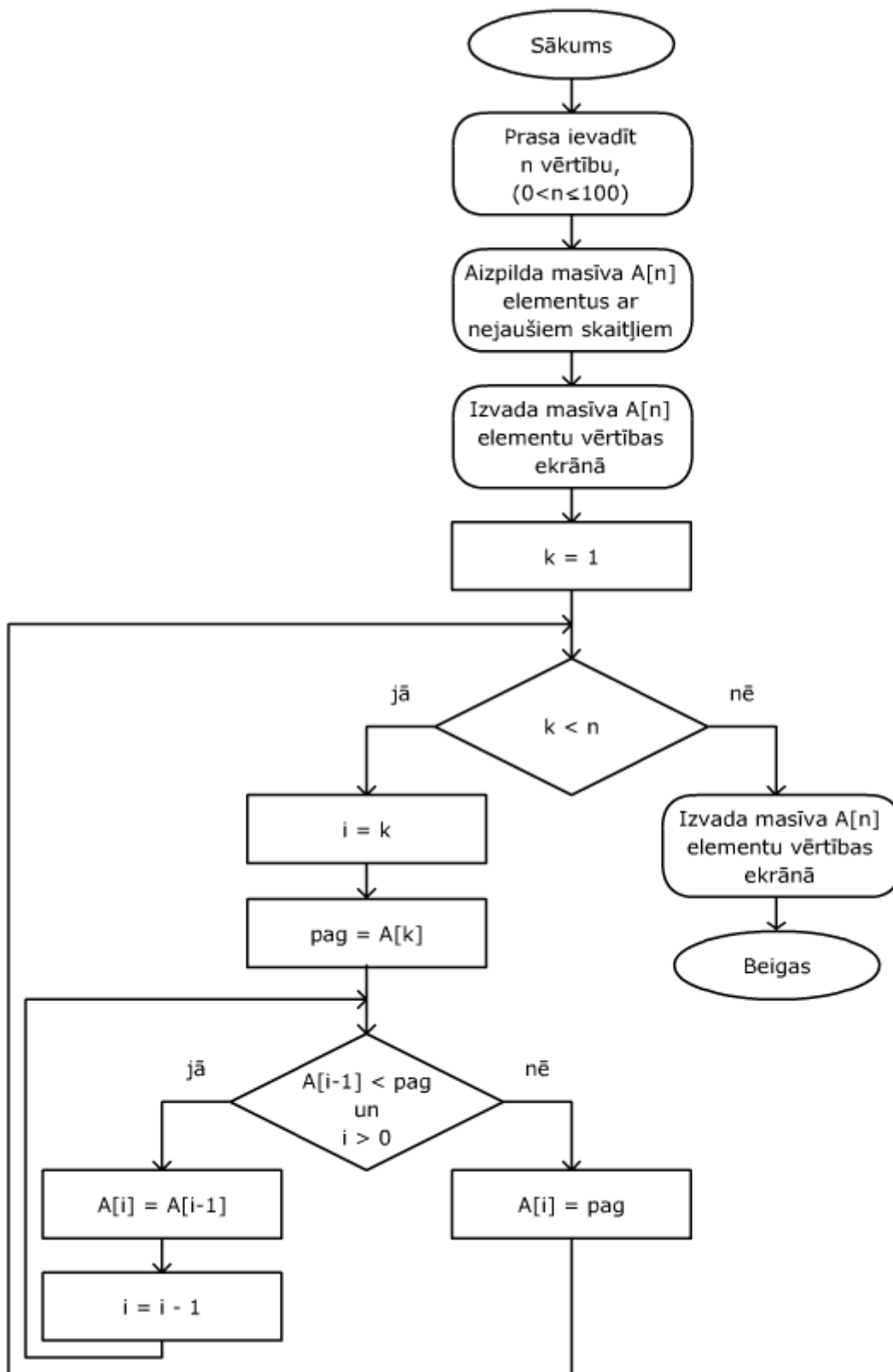
### 6. uzdevums [uzd6] (10 punkti)

Sastādi programmu, kas ekrānā izvada dotās tabulas vērtības.

0	1	2	3	4	5	6	7
1	0	1	2	3	4	5	6
1	2	0	1	2	3	4	5
1	2	3	0	1	2	3	4
1	2	3	4	0	1	2	3
1	2	3	4	5	0	1	2
1	2	3	4	5	6	0	1
1	2	3	4	5	6	7	0

### 7. uzdevums [uzd7] (10 punkti)

Sastādi programmu pēc dotās blokshēmas. Ņem vērā, ka masīva A pirmā elementa kārtas indekss ir nulle.



### 8. uzdevums [uzd8] (5 punkti)

Alise saņēma SMS ar tekstu "skeintsap vet tilut sisenta ilutsev." un izlasījusi to, neko nesaprata. Pēc brīža pie durvīm piezvanīja. Pastnieks bija atnesis viņai adresētu vēstuli, bet arī tās teksts bija nesaprotams. Meitene pārdomu pilna staigāja uz priekšu un atpakaļ pa istabu līdz ieraudzīja savu atspulgu spogulī un iesmējās, "Nu, protams, ...".

Sastādi programmu, kas palīdz Alisei izlasīt vēstules tekstu, kas ierakstīts teksta datnē *uzd8.in*. Rezultātu saglabā teksta datnē *uzd8.out*.

### 9. uzdevums [uzd9] (10 punkti)

Kādā karaļvalstī dzīvo pūķis – cilvēkēdājs. Viņam ir  $n$  galvas un  $m$  astes. Princis ir nolēmis "pielikt punktu" pūķa izdarībām un nokaut to. Taču to izdarīt nemaz nav tik vienkārši, jo,

- ja pūķim ar vienu zobena cirtienu nocērt vienu galvu, tās vietā ataug jauna;
- ja nocērt vienu asti, tās vietā ataug divas astes;
- ja nocērt divas astes, tās vietā ataug 1 galva;
- ja nocērt divas galvas, to vietā neataug nekas.

Sastādi programmu, kas nosaka, vai princim izdevās uzvarēt pūķi, ja teksta datnes *uzd9.in* pirmajā rindā doti trīs naturāli skaitļi, kas atdalīti ar atstarpēm:  $n$  – pūķa galvu skaits ( $0 < n \leq 100$ ),  $m$  – pūķa astu skaits ( $0 < m \leq 100$ ),  $k$  – prinča izdarīto cirtieņu skaits ( $0 < k \leq 100$ ), bet otrajā rindā dotas  $k$  komandas, kas atdalītas ar atstarpēm, kur G1 nozīmē, ka princim izdodas nocirst vienu pūķa galvu, G2 – 2 galvas, A1 – vienu asti un A2 – divas astes. Cīņas rezultātu izvadīt teksta datnē *uzd9.out* sekojošā veidā: ja pūķis uzvarēts, izvadīt tekstu "UZVARA"; ja cīņa zaudēta, izvadīt divus skaitļus, kas atdalīti ar atstarpi – pūķa galvu skaits un pūķa astu skaits cīņas beigās.

### 10. uzdevums [uzd10] (15 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.**

Palīdzī 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
```