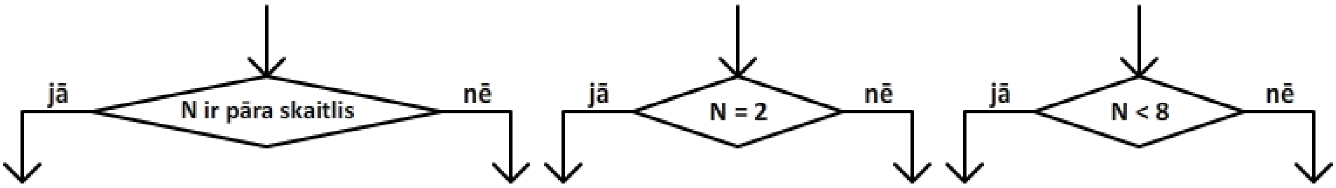
Liepājas Universitātes
Datorzinātņu olimpiādes 2017
programmēšanas konkursa uzdevumi studentiem

1. uzdevums [*uzd1*] (5 punkti)
Uzzīmēt blokshēmu, pēc kuras atbilstoši mēneša numuram N, varētu noteikt, cik dienas ir mēnesī. Blokshēmai jāsatur dotie blokshēmu elementi.



2. uzdevums [*uzd2*] (5 punkti)
Lieldienu Zaķa noliktava aprīkota ar īpašu digitālu slēdzeni, lai to atvērtu, jāievada ciparu kods – naturāls skaitlis N (1 ≤ N ≤ 10000). Ja ievadītais ciparu kods ir skaitlis, kas mazāks par 100 vai lielāks par 999, slēdzenes displejā parādās paziņojums "ERROR"; ja tiek ievadīts ciparu kods, kas veido skaitli, kura ciparu summa ir skaitlis 15, 16 vai 17, tad slēdzenes displejā parādās paziņojums "UNLOCK"; visos citos gadījumos parādās paziņojums "LOCK".
Sastādīt programmu, kas nosaka un izvada ekrānā slēdzenes displeja ekrānā redzamo paziņojumu, atbilstoši lietotāja ievadītajam ciparu kodam.

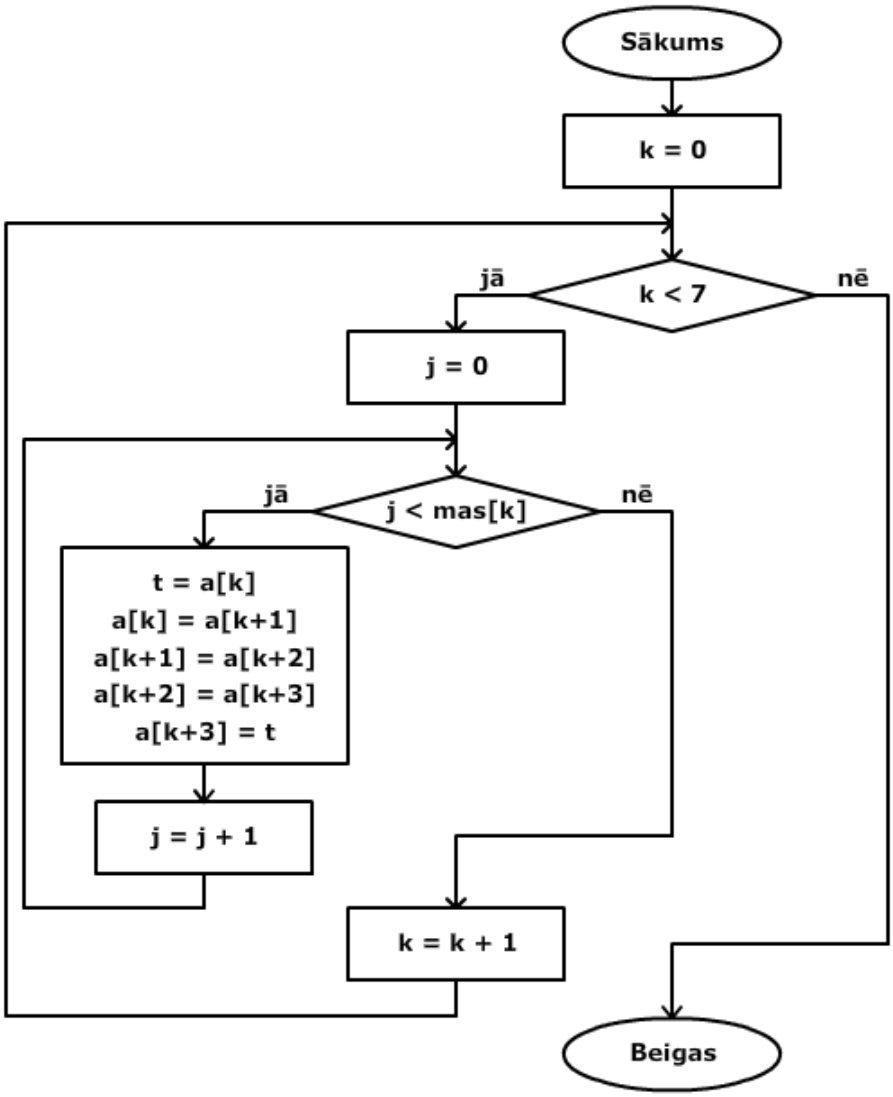
3. uzdevums [*uzd3*] (10 punkti)
Atrodi sakarības starp skaitļiem tabulas rūtiņas. Sastādi programmu, kas, izmantojot atrasto sakarību, ekrānā izvada dotās tabulas vērtības.

0	2	4	0	2	4	0	2	4
2	4	0	2	4	0	2	4	0
4	0	2	4	0	2	4	0	2
0	2	4	0	2	4	0	2	4
2	4	0	2	4	0	2	4	0
4	0	2	4	0	2	4	0	2
0	2	4	0	2	4	0	2	4
2	4	0	2	4	0	2	4	0
4	0	2	4	0	2	4	0	2

4. uzdevums [*uzd4*] (10 punkti)
Sastādīt programmu, kas no pirmskaitļu virknes, kuras pēdējā elementa vērtība nepārsniedz ievadīto naturālo skaitli N (1 ≤ N ≤ 1000), izdrukā ekrānā vidējā elementa vērtību, ja pirmskaitļu virknē ir nepāra skaits elementu, vai divu vidējo elementu vērtības – ja virknē ir pāra skaits elementu. Programmai jānodrošina, ka lietotājs nevar ievadīt nekorektu N vērtību.
Piemēram, ja lietotājs ievada 10, programma izdrukā ekrānā 3, bet, ja lietotājs ievada 25, programma izdrukā 7 un 11.

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

Dota algoritma blokshēma. Masīva ***a*** vērtības pirms algoritma izpildes ir {0, 1, 2, 3, 4, 5, 6, 7, 8, 9}, bet pēc izpildes {3, 0, 4, 5, 1, 2, 7, 8, 9, 6}. Pēc blokshēmas noteikt masīva ***mas*** elementu skaitu un vērtības, ja zināms, ka masīva ***mas*** vērtības ir veseli skaitļi intervālā [0;3].



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

Viesnīcā ierodas kaprīzs klients un vēlas apmesties istabā, kura atrodas vistuvāk liftam. Ja brīvas ir vairākas istabas, kuras atrodas vienādā attālumā no lifta, tad tajā, kura atrodas zemākā stāvā. Zināms, ka viesnīcas istabas kārtas numurs sastāv no diviem ciparіem, pirmais cipars apzīmē stāvu, kurā istaba atrodas, otrais norāda istabas attālumu no lifta. Viesnīcai ir 9 stāvi un katrā no tiem ir 9 istabas.

Sastādīt programmu, kas no faila *uzd6.in* nolasa brīvo viesnīcas istabu numurus (istabu numuri doti vienā rindā, tie atdalīti ar vienu atstarpi) un sarindo tos klientam vēlamajā secībā, sākumā viesnīcas istabas numurs, kurš visprecīzāk atbilst klienta vēlmēm, beigās istabas numurs, kurš vismazāk atbilst klienta vēlmēm. Sakārtotos viesnīcas istabu numurus izdrukāt failā *uzd6.out* (vienā rindā atdalītus ar atstarpi).

Piemēram,

ievaddati *uzd6.in*	izvaddati *uzd6.out*
12 25 11 43 22 15 18 31 44 52	11 31 12 22 52 43 44 15 25 18

7. uzdevums [*uzd7*] (15 punkti)

Lai reģistrētos sistēmā, lietotājam jāizvēlas droša parole. Par drošu paroli uzskata simbolu virkni, kura atbilst sekojošiem nosacījumiem:

1) Minimālais paroles garums - 8 simboli, maksimālais – 20 simboli;
2) Parole satur tikai ciparus un latīņu alfabēta lielos un mazos burtus;
3) Parolei jāsatur vismaz viens cipars, vismaz viens latīņu alfabēta lielais burts un vismaz viens latīņu alfabēta mazais burts;
4) Parole nedrīkst saturēt divus pēc kārtas vienādus burtus, tai skaitā, ja viens ir lielais, otrs mazais burts;
5) Parole nedrīkst saturēt trīs pēc kārtas vienādus ciparus.

Sastādīt programmu, kas nolasa iespējamo paroli (maksimālais garums 255 simboli) no faila *uzd7.in* un nosaka, vai parole ir droša vai nav. Ja parole ir droša, tad failā uzd7.out izvadīt "YES", ja nav droša, tad – "NO".

Piemēram,

ievaddati *uzd7.in*	izvaddati *uzd7.out*
Parole333eloraP	NO

ievaddati *uzd7.in*	izvaddati *uzd7.out*
Oola7aloO	NO

ievaddati *uzd7.in*	izvaddati *uzd7.out*
Zakis370	YES

8. uzdevums [*uzd8*] (10 punkti)

Ir 2500 gada sākums. Uz dzīvi Robotlandē pārcēlušies N jauni roboti. Cik roboti dzīvos Robotlandē K gada beigās, ja zināms, ka tie dzīvo pēc sekojošiem likumiem:

- katra gada sākumā visi roboti tiek sadalīti grupās pa 3 robotiem;
- viena gada laikā 3 robotu grupa uzbūvē 5 jaunus robotus;
- katra gada beigās robotus, kuri sasnieguši 3 gadu vecumu, izslēdz un nodod pārstrādei.

Sastādīt programmu, kas nolasa faila *uzd8.in* datus, aprēķina un izvada failā *uzd8.out* robotu skaitu K gada beigās. Faila *uzd8.in* pirmajā rindā doti divi naturāli skaitļi N ($1 \le N \le 12$) un K ($2500 \le K \le 2515$), kas atdalīti ar atstarpi.

Piemēram,

ievaddati *uzd8.in*	izvaddati *uzd8.out*
3 2503	115

9. uzdevums [*uzd9*] (15 punkti)

Olivers un Lūcija spēlē spēli "Uzmini, kurš burts paliks pēdējais?". Lūcija nosauc kādu iedomātu vārdu un jautā Oliveram, kurš burts vārdā paliks pēdējais. Spēles noteikumi - vārdā izsvītro katru otro burtu, tad atlikušo burtu virkni apgriež uz otru pusi, un atkal izsvītro katru otro burtu; šādi turpina, līdz atliek viens vienīgs burts. Piemēram, ja Lūcija nosauc vārdu "KARTUPELIS", tad pareizā atbilde ir "I", jo

KARTUPELIS ->

K ~~A~~ R ~~T~~ U ~~P~~ E ~~L~~ I ~~S~~ -> KRUEI -> IEURK ->

I ~~E~~ U ~~R~~ K -> IUK -> KUI ->

K ~~U~~ I -> KU -> IK ->

I ~~K~~ -> I.

Sastādīt programmu, kas nolasa failā *uzd9.in* dotos vārdus (katrs vārds dots savā rindā, maksimālais vārda garums 30 burti, vārds var saturēt gan lielos gan mazos latīņu alfabēta burtus, maksimālais vārdu skaits failā 100 vārdi) un failā *uzd9.out* katru savā rindā izdrukā no katra vārda pēdējo burtu, kas iegūts spēlējot "Uzmini, kurš burts paliks pēdējais?".

Piemēram,

ievaddati *uzd9.in*	izvaddati *uzd9.out*
KARTUPELIS	I
BUMBA	A
KURMIS	K
SUNS	N

10. uzdevums [*uzd10*] (10 punkti)

Naturālo skaitļus var sadalīti grupās, piemēram:

A. (1),(2,3),(4,5,6),(7,8,9,10),(11,12,13,14,15),...
B. (1),(2,3,4),(5,6,7,8,9),(10,11,12,13,14,15,16),...

Artūrs ir uzrakstījis programmu, kas nolasa no faila vienu skaitli N un aprēķina N-tās skaitļu grupas summu ($1 \le N \le 100$), ja skaitļi sadalīti grupās, kā parādīts piemērā A. Aprēķinātā skaitļu summa tiek izdrukāta otrā failā. Piemēram, ja no faila tiek nolasīts skaitli 5, tad programma otrā failā izdrukā skaitli 65.

Pārveido Artūra uzrakstīto programmu tā, lai ar tās palīdzību varētu aprēķināt N-tās skaitļu grupas summu, ja skaitļi sadalīti grupās, kā parādīts piemērā B.

(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);
   int N, M, i, j, SUM=0;
   dati1>>N;
   M=N*(N-1)/2;
   j=M;
   for(i=1; i<N+1; i++){
        j++;
        SUM +=j;
   }
   dati2<<SUM;
   dati1.close();
   dati2.close();
}
```

(Java kods)

```
import java.io.*;
public class Uzd10 {
  public static void main(String[] args) {
    try{
      BufferedReader dati1=new BufferedReader
                 (new FileReader("uzd10.in"));
      BufferedWriter dati2=new BufferedWriter
                 (new FileWriter("uzd10.out"));
      int N, M, i, j, SUM=0;
      N=Integer.parseInt(dati1.readLine());
      M=N*(N-1)/2;
      j=M;
      for(i=1; i<N+1; i++){
        j++;
        SUM +=j;
      }
      dati2.write(Integer.toString(SUM));
      dati1.close();
      dati2.close();
    } catch (IOException e){}
  }
}
```