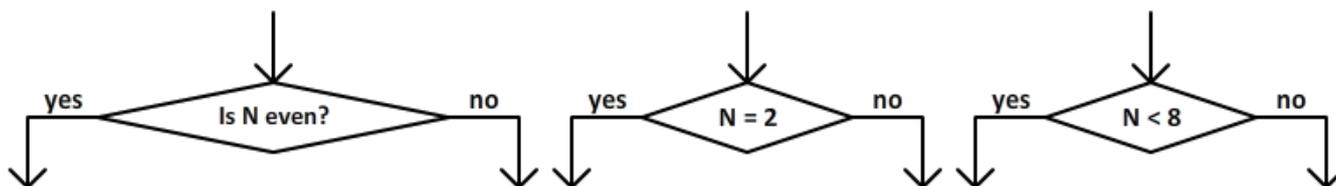


Liepājas Universitātes  
**Datorzinātņu olimpiādes 2017**  
 programmēšanas konkursa uzdevumi studentiem

**Task 1 [uzd1]** (5 points)

Draw the flowchart by that to detect how many days has the N-th month ( $N = 1, 2, 3, \dots, 12$ ). The flowchart should contain the given elements:



**Task 2 [uzd2]** (5 points)

Easter Bunny's warehouse equipped with special digital lock. In order to open it, the numeric code should be input, i.e., integer  $N$  ( $1 \leq N \leq 10000$ ). If the entered digital code is less than 100 or exceeds 999, the message "ERROR" is output on display of the lock. If sum of digits of the entered code is either 15, or 16, or 17 then the message "UNLOCK" is output on the display. The message "LOCK" is output in all other cases.

Write program that determine the message on the display of the digital lock according to entered digital code.

**Task 3 [uzd3]** (10 points)

Find the relationship between the figures of the table cells. Write the program that output the given values using the found relationship.

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

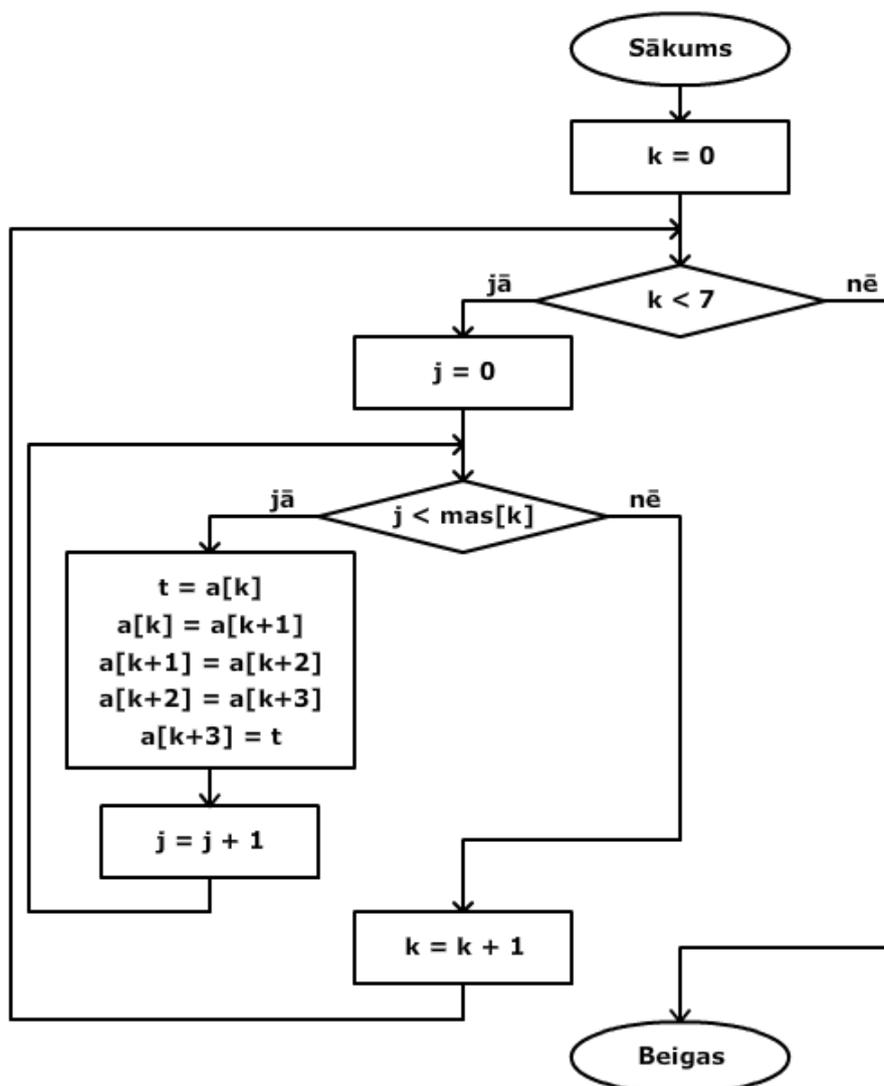
**Task 4 [uzd4]** (10 points)

Let us given the sequence of the all prime numbers not greater than  $N$  ( $1 \leq N \leq 1000$ ). Write the program that calculates either mean (*median*) element, if the number of elements is odd, or two mean elements, if the number of elements is even. The program must ensure that the user cannot enter the incorrect value of  $N$ .

*Example.* If input is 10, output is 3; if input is 25, output is 7 and 11.

### Task 5 [uzd5] (10 points)

Let us given the flowchart. Before the run of an algorithm, array **a** have the following values {0, 1, 2, 3, 4, 6, 7, 8, 9}. Before the run of an algorithm, array **a** have the following values {3, 0, 4, 5, 1, 2, 7, 8, 9, 6}. Reading the flowchart, determine the number of elements of array **mas** and its values, if the values of array **mas** are integers between 0 and 3 (including).



### Task 6 [uzd6] (10 points)

A cranky client arrives at the hotel. He wants to settle in the room that is located closest to the lift. In case of several vacant rooms located in similar distance from the lift, the client wants to occupy the room located in the lower floor. The number of the each room consists of two digits. The first digit identifies the floor, and the second one determines the distance from the lift. The hotel has 9 floors, each floor has 9 rooms.

Write the program that reads numbers of vacant rooms from text file *uzd6.in*; numbers of the rooms are written in one line separated by whitespace. The program must sort numbers in order according to requirements of the client: at first the room that fits best, followed by the worse fitted room, etc. The program must store the sorted order of rooms in text file *uzd6.out*; all in one line separated by whitespace.

Example,

Input data *uzd6.in*

12 25 11 43 22 15 18 31 44 52

Output data *uzd6.out*

11 31 12 22 52 43 44 15 25 18

**Task 7 [uzd7]** (15 points)

In order to sign in the system, user should create secure password, i.e. string of characters that follows the defined rules:

- 1) Length of password is between 8 and 20 (included);
- 2) Password includes only numbers and lowercase and uppercase alphabetic characters;
- 3) Password must contain at least one digit, at least one lowercase alphabetic character and, and at least one uppercase alphabetic character;
- 4) Two adjacent equal characters are prohibited; even if one of them is uppercase and another one is lowercase;
- 5) Three adjacent equal digits are prohibited.

Write program that read the potential password (max. 255 characters) from text file *uzd7.in* and determines if it is secure password. In case of secure password the program writes "YES" in text file *uzd7.out*, otherwise the output is "NO".

Example,

Input data *uzd7.in*  
Parole333eloraP

Output data *uzd7.out*  
NO

Input data *uzd7.in*  
Oola7aloO

Output data *uzd7.out*  
NO

Input data *uzd7.in*  
Zakis370

Output data *uzd7.out*  
YES

**Task 8 [uzd8]** (10 points)

At the beginning of Year 2500, N new robots has moved to Robotland. How many robots will live in Robotland at the end of Year K, if they are follows the rules:

- At the beginnig of the each year, all robots are grouped by three in a team;
- During the year, each team of three robots build five new robots;
- At the end of year, the 3-years-old robots are switched off and recycled.

Write program that reads data from text file *uzd8.in*, claculates number of robots at the end of Year K, and store the result in text file *uzd8.out*. The first line of the text file *uzd8.in* contains two integers N ( $1 \leq N \leq 12$ ) and K ( $2500 \leq K \leq 2515$ ) separated by whitespace.

Example,

Input data *uzd8.in*  
3 2503

Output data *uzd8.out*  
115

**Task 9 [uzd9]** (15 points)

Oliver and Lucy play game "Guess: which letter will be the last one?". Lucy names a guessed word and asks Oliver, which letter will be the last one. Rules of the games are the following: (a) at first, remove every second letter, (b) then reverse the string of remaining letters (turn in backward order), (c) repeat steps a and b until single letter left. For example, when Lucy names word "KARTUPELIS", the wright answer is "I", because

```
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.
```

Write program that reads data from text file uzd9.in the given words; each word is written in separate line; maximum length of the word is 30 letters; word can include both uppercase, and lowercase Latin alphabet letters; the file contains no more than 100 words. The program must write the last remained letter for each word after the gameplay in text file uzd9.out; each result in separate line.

Example,

Input data <i>uzd9.in</i>	Output data <i>uzd9.out</i>
KARTUPELIS	I
BUMBA	A
KURMIS	K
SUNS	N

**Task 10 [uzd10]** (10 points)

Integers could be divided in several groups, for example:

- 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),...

Arthur has written program (see below) that reads an integer N from text file, calculates sum of numbers included in N-th group ( $1 \leq N \leq 100$ ); and result must be written into another text file. The groups have created by the rule of case A. For example, when input data is 5, the output is 65. Modify the Arthur's program, let it do the same for the case B.

**(C++ code)**

```
#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 code)**

```
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){}
    }
}
```