

Liepājas Universitātes
Datorzinātņu olimpiādes 2018
 programmēšanas konkursa uzdevumi studentiem

Task 1 [uzd1] (5 points)

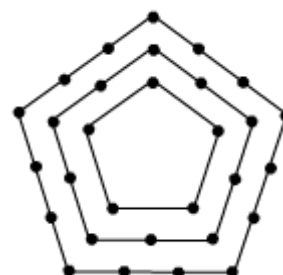
On January 1st, 2500, N new robots have moved to Robotown. They are going to live by the following rules:

- At the beginning of every year, all robots are assigned to teams by 4 robots in each team;
- Within a year, each team (4 robots) build 5 new robots;
- Every year on December 31st, 3-years old robots are turned off and handed over for recycling.

Draw a flowchart by which to determine number of robots in Robotown after K years.

Task 2 [uzd2] (5 points)

The tower has K floors. Each floor has a regular N-corner shape. On the top floor of the tower, there are two hunting hatches in each wall, i.e., one hatch in each corner. On each wall of the next bottom floor, there are 1 more hatches more than the wall of the previous (upper) floor. Write a program that determines number of hunting hatches in the tower and output it on the display screen. User must input the K and N values from the keyboard. Your program must ensure that the user can input only correct values: $1 \leq K \leq 1000$ and $3 \leq N \leq 1000$.



For example: The figure represents the view on the tower from the top that has 3 floors and each floor has pentagon shape.

Task3 [uzd3] (10 points)

Find relation between numbers in the given table that has N rows and M columns. Write a program that output similar table (without cell border lines) using the found relationship. User must input the N and M values from the keyboard. Your program must ensure that the user can input only correct values: N is multiple of 3, $3 \leq N \leq 21$, and M is even number, $2 \leq M \leq 20$.

For example: For N=9 and M=8, the given table looks like the following:

1	1	2	2	3	3	4	4
5	1	6	2	7	3	8	4
5	5	6	6	7	7	8	8
9	9	10	10	11	11	12	12
13	9	14	10	15	11	16	12
13	13	14	14	15	15	16	16
17	17	18	18	19	19	20	20
21	17	22	18	23	19	24	20
21	21	22	22	23	23	24	24

Task 4 [uzd4] (10 points)

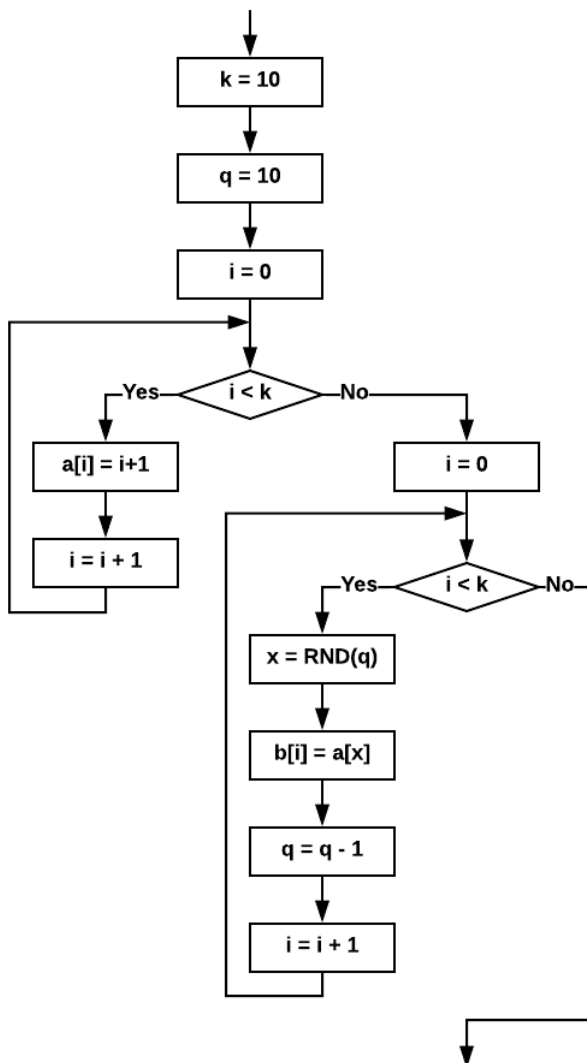
Write a program that outputs a correct algebraic expression $a+bx+cy$ for any values of a, b, and c. User must input the a, b, and c values from the keyboard.

For example: For $a=2$, $b=0$, and $c=3$, the program must output $2+3y$; for $a=3$, $b=-1$, and $c=0$, the program must output $3-x$

Task 5 [uzd5] (10 points)

See the given part of a flowchart. RND(q) devotes a function that returns integer n, $0 \leq n < q$. Reading flowchart determines:

- a) Determines number of items in array a;
- b) Determines number of items in array b;
- c) Determines the values of items in array a;
- d) Determines the values of items in array b;
- e) Describe problem solved by the given flowchart.



Task 6 [uzd6] (10 points)

A cranky customer arrives at the hotel. He wants to stay in the room that is closest to the elevator. If there are several rooms available that are equidistant from the elevator, then customer selects the room on the lower floor. The room identifier is a two-digit number – the first digit indicates the floor where the room is located and the second digit indicates the distance from the elevator. The hotel has 9 floors, there are 9 rooms located in the each floor.

Write a program that reads from text file *uzd6.in* identifiers of available rooms (all identifiers have written in one row separated by whitespace character) and sorts them according to customer’s priorities, i.e., the first identifier matches most closely to customer’s request, and the last one matches least. Your program must write the result into text file *uzd6.out* – all in one row, each identifier separated by whitespace character.

For 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)

A chessboard is checked board consisting of 8x8 black and white squares. Each square has its own coordinate, defined by one capital letter and one digit. Each of the chess pieces moves differently. So, the knight moves to a square that is two squares away horizontally and one square vertically, or two squares vertically and one square horizontally.

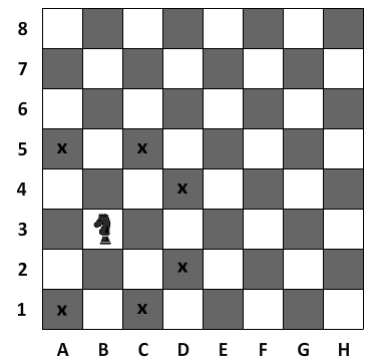
Write a program that reads from the single line of the text file *uzd7.in* starting coordinates of the knight, determines the colour of the square under the knight and coordinates of all possible squares where the knight can jump after one move.

Your program must write the result into text file *uzd7.out*. The first line should contain a word either BALTS, if the knight starts the move from white square, or MELNS, if the knight starts the move from black square. The next lines must contain coordinates of all possible squares where the knight can jump after the one move; each line must contain only coordinates of one square.

For example (see figure):

Input data *uzd7.in*:
B3

Output data *uzd7.out*:
BALTS
A5
C5
D4
D2
C1
A1



Task 8 [uzd8] (10 points)

Treasure hunters, guided by an old pirates' map, have caught up on the uninhabited island. They have put up a camp there. Early in the morning, following the pirate's instructions, the treasure hunters go for the hidden treasures. After a long trip breaking through the overgrown jungles and crossing the marshes finally they get to the treasures in late afternoon. Because the treasure chest is very heavy and the treasure hunters are very tired, they decide to return to the camp by the shortest way.

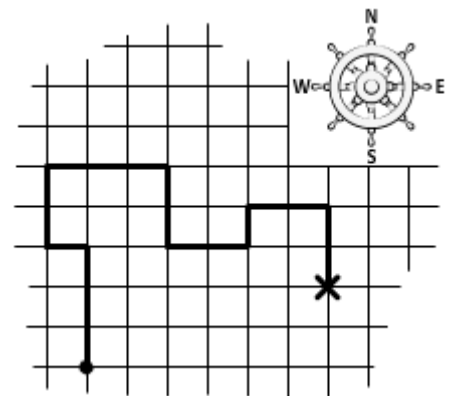
Write a program that reads from text file *uzd8.in* pirates' instructions, and finds the distance between the treasures and the camp. Each line of the input file has one instruction defined by a capital letter and an integer X , $1 \leq X \leq 1000$ separated by a whitespace character. The capital letter defines the direction of the movement, i.e., N – North, E – East, S – South, W – West. The integer X defines the number of steps.

Your program must write the result into text file *uzd8.out*

For example (see figure):

Input data *uzd8.in*
N3
W1
N2
E3
S2
E2
N1
E2
S2

Output data *uzd8.out*
6.324555



Task 9 [uzd9] (15 points)

The single line of the text file *uzd9.in* contains an even integer that has at least 2 digits but up to 100000 digits. Write a program that resets the digits of the given number by the following rules:

- At first, all digits are divided into two equal parts;
- After that, the sequence of digits of each part are reversed;
- Then repeat the step (a) extracting the first digit and the last one of the obtained number;
- Then repeat the step (a) extracting the first two digits and the last two digits of the obtained number;
- Repeat the step (a) increasing each time the number of extracted digits until number of reversible digit is 1.

Your program must write the obtained number into text file *uzd9.out*

For example:

The text file *uzd9.in* has an integer 1234554321

The following steps have been completed:

```
12345|54321
54321|12345
51234|43215
51432|23415
51423|32415
```

The result written in the text file *uzd9.out* is 5142332415

Task 10 [uzd10] (10 points)

Kurts has been learning the basics of programming at school. During the class, Kurts has written a program that calculates the sum of two large integers.

Modify the program written by Kurts. It must read two large integers from the text file *uzd10.in*, each integer has been written in its own line. The program must calculate the difference between the input numbers and write the result in the text file *uzd10.out*.

C++ code for the calculation of the sum of two large integers

```
#include <fstream>
using namespace std;
int main(){
    ifstream dati1;
    ofstream dati2;
    dati1.open("uzd10.in",ios::in);
    dati2.open("uzd10.out",ios::out);
    string sk1, sk2;
    int n1, n2, n3;
    int i;
    dati1>>sk1;
    dati1>>sk2;
    n1=sk1.length();
    n2=sk2.length();
    if(n1>n2) n3=n1; else n3=n2;
    int x1[n3+1]={0};
    int x2[n3+1]={0};
    int x3[n3+1]={0};
    for(i=0; i<n1; i++) x1[i]=sk1[n1-i-1]-'0';
    for(i=0; i<n2; i++) x2[i]=sk2[n2-i-1]-'0';
    for(i=0; i<n3; i++){
        x3[i+1]=(x1[i]+x2[i]+x3[i])/10;
        x3[i]=(x1[i]+x2[i]+x3[i])%10;
    }
    if(x3[n3]!=0) dati2<<x3[i];
    for(i=n3-1; i>=0; i--) dati2<<x3[i];
    dati1.close();
    dati2.close();
}
```

Java code for the calculation of the sum of two large integers

```
import java.io.*;
public class Sum {
    public static void main(String[] args) {
        try{
            BufferedReader dati1=new BufferedReader(new FileReader("uzd10.in"));
            BufferedWriter dati2=new BufferedWriter(new FileWriter("uzd10.out"));
            String sk1, sk2;
            int n1, n2, n3;
            int i;
            sk1=dati1.readLine();
            sk2=dati1.readLine();
            n1=sk1.length();
            n2=sk2.length();
            n3=Math.max(n1,n2);
            int[] x1=new int[n3+1];
            int[] x2=new int[n3+1];
            int[] x3=new int[n3+1];
            for(i=0; i<n1; i++) x1[i]=sk1.charAt(n1-i-1)-'0';
            for(i=0; i<n2; i++) x2[i]=sk2.charAt(n2-i-1)-'0';
            for(i=0; i<n3; i++){
                x3[i+1]=(x1[i]+x2[i]+x3[i])/10;
                x3[i]=(x1[i]+x2[i]+x3[i])%10;
            }
            if(x3[n3]!=0) dati2.write(x3[i]+"");
            for(i=n3-1; i>=0; i--) dati2.write(x3[i]+"");
            dati1.close();
            dati2.close();
        } catch (IOException e){System.out.print("Nevar atrast failu!");}
    }
}
```

Python3 code for the calculation of the sum of two large integers

```
dati1 = open("uzd10.in", "r")
dati2 = open("uzd10.out", "w")
n1=0; n2=0; n3=0
i=0
sk1 = dati1.readline()
sk2 = dati1.readline()
n1=len(sk1)-1
n2=len(sk2)
if n1>n2: n3=n1
else: n3=n2
x1 = [0] * (n3+1)
x2 = [0] * (n3+1)
x3 = [0] * (n3+1)
for i in range(n1): x1[i]=int(sk1[n1-i-1])
for i in range(n2): x2[i]=int(sk2[n2-i-1])
for i in range(n3):
    x3[i+1]=int((x1[i]+x2[i]+x3[i])/10)
    x3[i]=(x1[i]+x2[i]+x3[i])%10
if x3[n3] != 0: dati2.write(str(int(x3[i])))
for i in range(n3-1, -1, -1): dati2.write(str(int(x3[i])))
dati1.close()
dati2.close()
```