УО «Белорусский государственный университет информатики и радиоэлектроники»

Кафедра ПОИТ

Отчет по лабораторной работе №2.1

по предмету «Основы алгоритмизации и программирования»

Вариант 10

Выполнил:

Мухин Д.С.

Гр. 351004

Проверил:

Данилова Г. В.

Минск 2023

**Задание:**

Если последнюю цифру некоторого натурального числа n перенести и поставить перед первой цифрой этого числа, то получится число, в два раза больше n. Найти самое маленькое из таких чисел.

**Код программы Delphi:**

Program Lab2\_1;

Uses

SysUtils;

Const

MINSIDES = 3;

MAXSIDES = 100;

MAX\_VALUE = 1000;

MIN\_VALUE = -1000;

TWO = 2;

Var

Cross, Slope1, Slope2: Double;

SidesNumb, I, J: Integer;

IsCorrect, IsPositive, IsNegative, IsConvex: Boolean;

Matrix: Array Of Array Of Real;

Begin

Cross := 0.0;

Slope1 := 0.0;

Slope2 := 0.0;

SidesNumb := 0;

IsCorrect := True;

IsPositive := False;

IsNegative := False;

IsConvex := True;

Writeln('Данная программа проверяет многоугольник на выпуклость.');

Writeln('Количество вершин многоугольника должно быть от ', MINSIDES,

' до ', MAXSIDES, '.');

Write('Введите количество вершин многоугольника: ');

Repeat

IsCorrect := True;

Try

Readln(SidesNumb);

Except

Writeln('Проверьте корректность ввода данных!');

IsCorrect := False;

End;

If IsCorrect And ((SidesNumb < MINSIDES) Or (SidesNumb > MAXSIDES)) Then

Begin

Write('Некорректный ввод! Количество вершин многоугольника должно быть от ',

MINSIDES, ' до ', MAXSIDES, '.');

Writeln(' Повторите попытку!');

IsCorrect := False;

End;

Until IsCorrect;

Writeln('X и Y должны быть в диапазоне от ', MIN\_VALUE, ' до ', MAX\_VALUE);

SetLength(Matrix, SidesNumb, 2);

Repeat

For I := Low(Matrix) To High(Matrix) Do

Begin

IsCorrect := True;

Repeat

IsCorrect := True;

Repeat

IsCorrect := True;

Write('Введите x', I + 1, ': ');

Try

Readln(Matrix[I][0]);

Except

Writeln('Проверьте корректность ввода данных!');

IsCorrect := False;

End;

If IsCorrect And ((Matrix[I][0] < MIN\_VALUE) Or (Matrix[I][0] > MAX\_VALUE))

Then

Begin

Write('X должен быть в диапазоне от ', MIN\_VALUE,

' до ', MAX\_VALUE, '.');

Writeln(' Повторите попытку!');

IsCorrect := False;

End;

Until IsCorrect;

Repeat

IsCorrect := true;

Write('Введите y', I + 1, ': ');

Try

Readln(Matrix[I][1]);

Except

Writeln('Проверьте корректность ввода данных!');

IsCorrect := False;

End;

If (IsCorrect) And

((Matrix[I][1] < MIN\_VALUE) Or (Matrix[I][1] > MAX\_VALUE)) Then

Begin

Writeln('Проверьте корректность ввода данных!');

Write('Y должен быть в диапазоне от ', MIN\_VALUE,

' до ', MAX\_VALUE, '.');

Writeln(' Повторите попытку!');

IsCorrect := False;

End;

Until IsCorrect;

If (I > 1) And (Matrix[I - 1][0] - Matrix[I - 2][0] <> 0) Then

Begin

Slope1 := (Matrix[I - 1][1] - Matrix[I - 2][1]) /

(Matrix[I - 1][0] - Matrix[I - 2][0]);

Slope2 := Matrix[I - 1][1] - Matrix[I - 1][0] \* Slope1;

If (Matrix[I][1] = Slope1 \* Matrix[I][0] + Slope2) Then

Begin

Writeln('Три точки не должны лежать на одной прямой!');

Writeln('Введите координаты точки ', I + 1,

' повторно.');

IsCorrect := False;

End;

End

Else If (I > 1) And (Matrix[I][0] = Matrix[I - 1][0]) And

(Matrix[I][0] = Matrix[I - 2][0]) Then

Begin

Writeln('Три точки не должны лежать на одной прямой!');

Writeln('Введите координаты точки ', I + 1, ' повторно.');

IsCorrect := False;

End;

Until IsCorrect;

End;

For I := Low(Matrix) To High(Matrix) Do

Begin

For J := I + 1 To High(Matrix) Do

Begin

If (Matrix[I][0] = Matrix[J][0]) And

(Matrix[I][1] = Matrix[J][1]) Then

IsCorrect := False;

End;

End;

If IsCorrect <> True Then

Writeln('Точки не должны совпадать! Повторите попытку.');

Until IsCorrect;

For I := TWO To High(Matrix) Do

Begin

Cross := (Matrix[I - 1][0] - Matrix[I - 2][0]) \*

(Matrix[I][1] - Matrix[I - 1][1]) - (Matrix[I][0] - Matrix[I - 1][0])

\* (Matrix[I - 1][1] - Matrix[I - 2][1]);

If Cross > 0 Then

IsPositive := True

Else If Cross < 0 Then

IsNegative := True;

End;

If IsPositive And IsNegative Then

IsConvex := False

Else

IsConvex := True;

If IsConvex Then

Writeln('Многоугольник выпуклый.')

Else

Writeln('Многоугольник не выпуклый.');

Readln;

End.

**Код программы С++:**

#include <iostream>

const int MINSIDES = 3;

const int MAXSIDES = 100;

const int MAX\_VALUE = 1000;

const int MIN\_VALUE = -1000;

const int TWO = 2;

int main()

{

setlocale(LC\_ALL, "Russian");

double cross, slope1, slope2;

int sidesNumb, i;

bool isIncorrect, isPositive, isNegative, isConvex;

cross = 0.0;

sidesNumb = 0;

isIncorrect = true;

isPositive = false;

isNegative = false;

isConvex = true;

std::cout << "Данная программа проверяет многоугольник на выпуклость.\n";

std::cout << "Количество вершин многоугольника должно быть от " << MINSIDES << " до " <<

MAXSIDES << "\n";

do

{

std::cout << "Введите количество вершин многоугольника: ";

std::cin >> sidesNumb;

if (std::cin.get() != '\n')

{

std::cin.clear();

std::cin.ignore(30000, '\n');

std::cout << "Некорректный ввод! Повторите попытку.\n";

}

else if (sidesNumb < MINSIDES || sidesNumb > MAXSIDES)

std::cout << "Некорректный ввод! Количество вершин многоугольника должно быть от "

<< MINSIDES << " до " << MAXSIDES << "\n";

else

isIncorrect = false;

} while (isIncorrect);

float\*\* matrix = new float\* [sidesNumb];

std::cout << "X и Y должны быть в диапазоне от " << MIN\_VALUE << " до " << MAX\_VALUE <<

"\n";

do

{

for (i = 0; i < sidesNumb; i++)

{

matrix[i] = new float[2];

do

{

do

{

isIncorrect = false;

std::cout << "Введите x" << i + 1 <<": ";

std::cin >> matrix[i][0];

if (std::cin.fail() || std::cin.get() != '\n')

{

std::cin.clear();

std::cin.ignore(30000, '\n');

isIncorrect = true;

}

if (isIncorrect || matrix[i][0] < MIN\_VALUE || matrix[i][0] > MAX\_VALUE ||

(matrix[i][0] == 0 && std::signbit(matrix[i][0])))

{

isIncorrect = true;

std::cout << "Некорректный ввод! Повторите попытку.\n";

std::cout << "X должен быть в диапазоне от " << MIN\_VALUE << " до "

<< MAX\_VALUE << "\n";

}

else

isIncorrect = false;

} while (isIncorrect);

isIncorrect = true;

do

{

isIncorrect = false;

std::cout << "Введите y" << i + 1 << ": ";

std::cin >> matrix[i][1];

if (std::cin.fail() || std::cin.get() != '\n')

{

std::cin.clear();

std::cin.ignore(30000, '\n');

isIncorrect = true;

}

if (isIncorrect || matrix[i][1] < MIN\_VALUE || matrix[i][1] > MAX\_VALUE ||

(matrix[i][1] == 0 && std::signbit(matrix[i][1])))

{

isIncorrect = true;

std::cout << "Некорректный ввод! Повторите попытку.\n";

std::cout << "Y должен быть в диапазоне от " << MIN\_VALUE << " до "

<< MAX\_VALUE << "\n";

}

else

isIncorrect = false;

} while (isIncorrect);

isIncorrect = true;

if (i > 1 && matrix[i - 1][0] - matrix[i - 2][0] != 0)

{

slope1 = (matrix[i - 1][1] - matrix[i - 2][1]) / (matrix[i - 1][0] –

matrix[i - 2][0]);

slope2 = matrix[i - 1][1] - matrix[i - 1][0] \* slope1;

if (matrix[i][1] == slope1 \* matrix[i][0] + slope2)

std::cout << "Три точки не должны лежать на одной прямой!\n"

<< "Введите координаты точки" << i + 1 << " повторно\n";

else

isIncorrect = false;

}

else if (i > 1 && matrix[i][0] == matrix[i - 1][0]

&& matrix[i][0] == matrix[i- 2][0])

std::cout << "Три точки не должны лежать на одной прямой!\n"

<< "Введите координаты точки" << i + 1 << " повторно\n";

else

isIncorrect = false;

} while (isIncorrect);

}

for (int i = 0; i < sidesNumb; i++)

{

for (int j = i + 1; j < sidesNumb; j++)

{

if (matrix[i][0] == matrix[j][0] && matrix[i][1] == matrix[j][1])

isIncorrect = true;

}

}

if (isIncorrect)

std::cout << "Точки не должны совпадать! Повторите попытку\n";

} while (isIncorrect);

for (int i = 2; i < sidesNumb; ++i)

{

cross = (matrix[i - 1][0] - matrix[i - 2][0]) \* (matrix[i][1] - matrix[i - 1][1]) -

(matrix[i][0] - matrix[i - 1][0]) \* (matrix[i - 1][1] - matrix[i - 2][1]);

if (cross > 0)

{

isPositive = true;

}

else if (cross < 0)

{

isNegative = true;

}

}

if (isPositive && isNegative)

{

isConvex = false;

}

isConvex ? std::cout << "Многоугольник выпуклый.\n" : std::cout << "Многоугольник не

выпуклый.\n";

for (int i = 0; i < sidesNumb; i++)

{

delete[] matrix[i];

}

delete[] matrix;

return 0;

}

**Код программы Java:**

import java.util.Scanner;  
  
public class Lab2\_1 {  
  
 static final int MINSIDES = 3;  
 static final int MAXSIDES = 100;  
 static final int MAX\_VALUE = 1000;  
 static final int MIN\_VALUE = -1000;

static final int TWO = 2;  
  
 public static void main(String[] args) {  
 Scanner scan = new Scanner(System.in);  
  
 double cross, slope1, slope2;  
 int sidesNumb;  
 boolean isIncorrect, isPositive, isNegative, isConvex;  
  
 cross = 0.0;  
 sidesNumb = 0;  
 isIncorrect = true;  
 isPositive = false;  
 isNegative = false;  
 isConvex = true;  
  
 System.out.println("Данная программа проверяет многоугольник на выпуклость.");  
 System.out.println("Количество вершин многоугольника должно быть от " + MINSIDES + " до " + MAXSIDES);  
 System.out.print("Введите количество вершин многоугольника: ");  
  
  
 do {  
 isIncorrect = false;  
  
 try {  
 sidesNumb = Integer.parseInt(scan.nextLine());  
 } catch (NumberFormatException error) {  
 System.err.println("Проверьте корректность ввода данных!");  
 isIncorrect = true;  
 }  
  
 if (!isIncorrect && (sidesNumb < MINSIDES || sidesNumb > MAXSIDES)) {  
 System.err.println("Проверьте корректность ввода данных!");  
 System.err.println("Некорректный ввод! Количество вершин многоугольника должно

быть от " + MINSIDES + " до " + MAXSIDES + "\nПовторите

попытку!");   
 isIncorrect = true;  
 }  
  
 } while (isIncorrect);  
  
  
 System.out.println("X и Y должны быть в диапазоне от " + MIN\_VALUE + " до " +

MAX\_VALUE);

float[][] matrix = new float[sidesNumb][2];

do {

for (int i = 0; i < sidesNumb; i++) {

do {

do {  
  
 isIncorrect = false;  
  
 System.out.print("Введите x" + (i + 1) + ": ");  
  
 try {  
 matrix[i][0] = Float.parseFloat(scan.nextLine());  
 } catch (NumberFormatException error) {  
 System.err.println("Проверьте корректность ввода данных!");  
 isIncorrect = true;  
 }  
  
 if (!isIncorrect && (matrix[i][0] < MIN\_VALUE

|| matrix[i][0] > MAX\_VALUE || (matrix[i][0] == 0

&& Math.copySign(1, matrix[i][0]) == -1))){  
 System.err.println("Проверьте корректность ввода данных!");  
 System.err.println("X должен быть в диапазоне от " + MIN\_VALUE + "

до " + MAX\_VALUE + "\nПовторите попытку!");   
 isIncorrect = true;  
 }  
  
 } while (isIncorrect);  
  
  
 do {  
 isIncorrect = false;  
 System.out.print("Введите y" + (i + 1) + ": ");  
  
 try {  
 matrix[i][1] = Float.parseFloat(scan.nextLine());  
 } catch (NumberFormatException error) {  
 System.err.println("Проверьте корректность ввода данных!");  
 isIncorrect = true;  
 }  
  
 if (!isIncorrect && (matrix[i][1] < MIN\_VALUE

|| matrix[i][1] > MAX\_VALUE || (matrix[i][1] == 0

&& Math.copySign(1, matrix[i][1]) == -1))) {  
 System.err.println("Проверьте корректность ввода данных!");  
 System.err.println("Y должен быть в диапазоне от " + MIN\_VALUE + "

до " + MAX\_VALUE + "\nПовторите попытку!");  
 isIncorrect = true;  
 }  
  
 } while (isIncorrect);  
  
 isIncorrect = true;  
  
 if (i > 1 && matrix[i - 1][0] - matrix[i - 2][0] != 0) {  
 slope1 = (matrix[i - 1][1] - matrix[i - 2][1]) / (matrix[i - 1][0] –

matrix[i - 2][0]);   
 slope2 = matrix[i - 1][1] - matrix[i - 1][0] \* slope1;

if (matrix[i][1] == slope1 \* matrix[i][0] + slope2) {  
 System.err.println("Три точки не должны лежать на одной прямой!");  
 System.err.println("Введите координаты точки " + (i + 1) + "

повторно.");

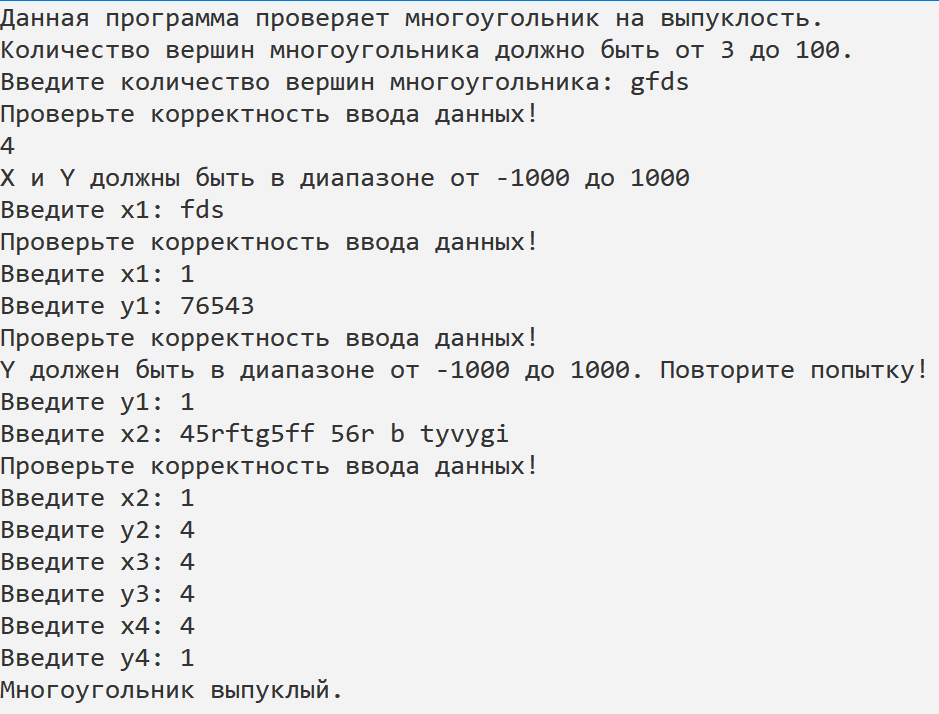
} else {  
 isIncorrect = false;  
 }  
 } else if (i > 1 && matrix[i][0] == matrix[i - 1][0]

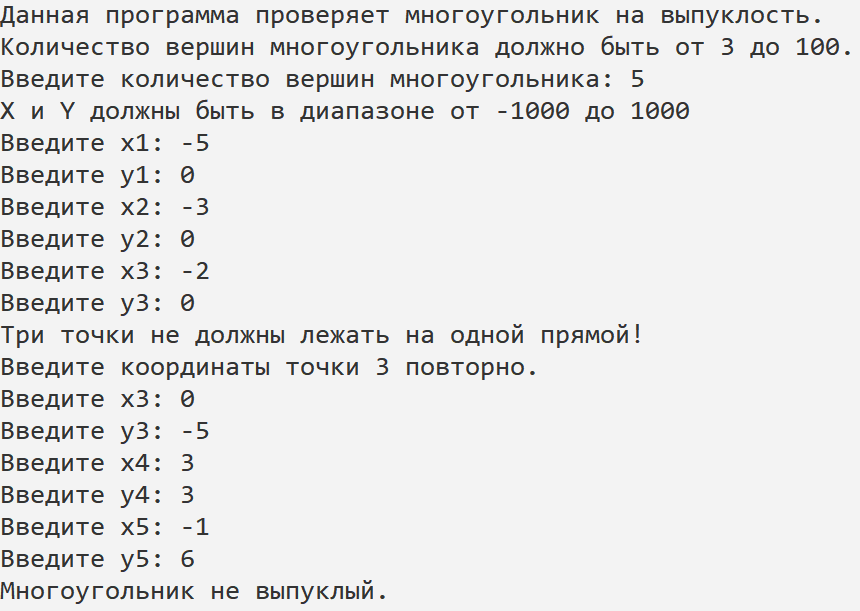
&& matrix[i][0] == matrix[i - 2][0]) {   
 System.err.println("Три точки не должны лежать на одной прямой!");  
 System.err.println("Введите координаты точки " + (i + 1) + "

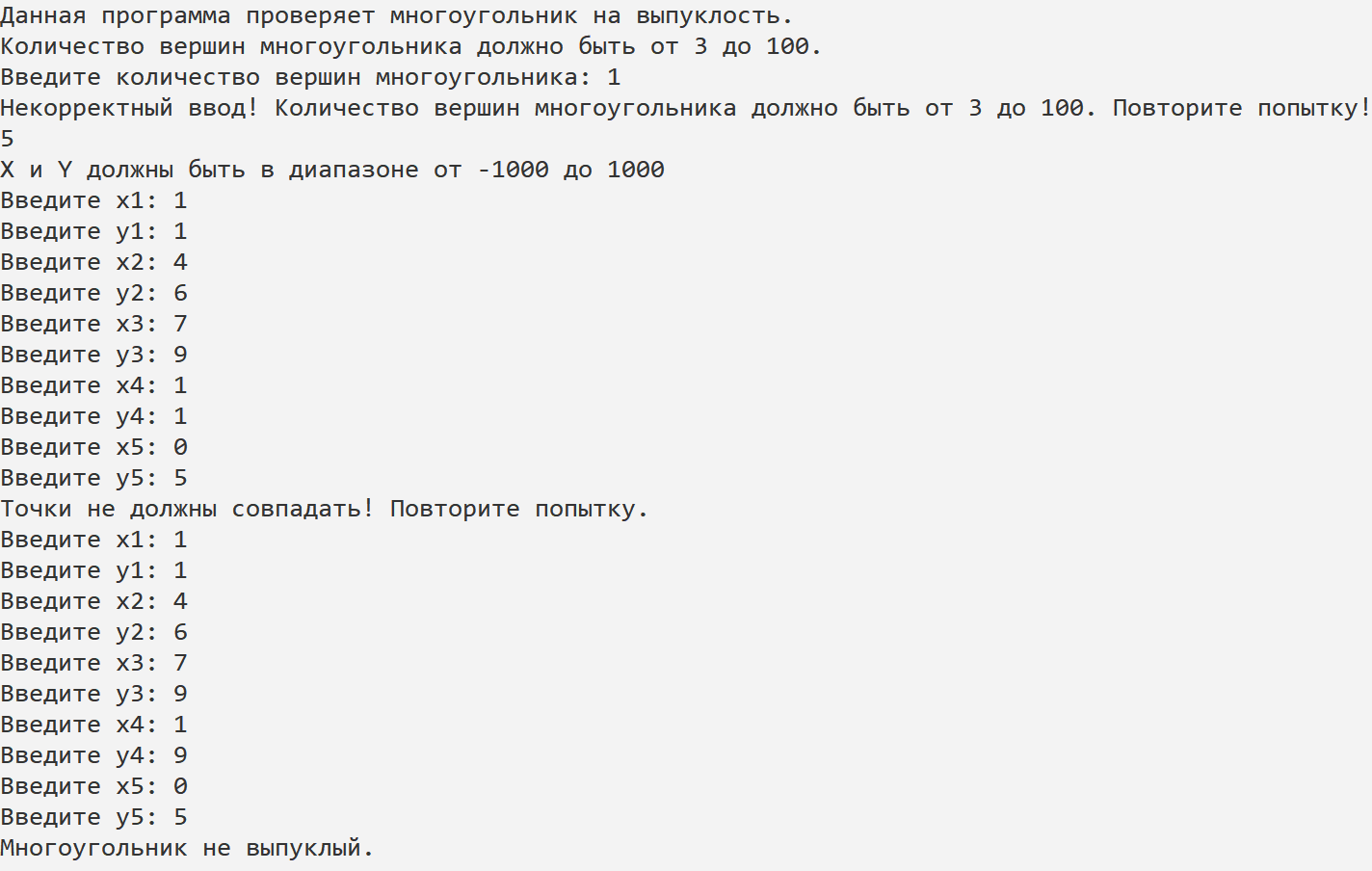
повторно.");   
 } else {  
 isIncorrect = false;  
 }  
  
  
 } while (isIncorrect);  
  
 }  
 for (int i = 0; i < sidesNumb; i++) {  
 for (int j = i + 1; j < sidesNumb; j++) {  
 if (matrix[i][0] == matrix[j][0] && matrix[i][1] == matrix[j][1]) {  
 isIncorrect = true;  
 }  
 }  
 }  
  
 if (isIncorrect) {  
 System.err.print("Точки не должны совпадать! Повторите попытку.\n");  
 }  
  
 }while (isIncorrect) ;  
  
  
  
 for (int i = 2; i < sidesNumb; ++i) {  
 cross = (matrix[i - 1][0] - matrix[i - 2][0]) \* (matrix[i][1] - matrix[i - 1][1]) -  
 (matrix[i][0] - matrix[i - 1][0]) \* (matrix[i - 1][1] - matrix[i - 2][1]);  
  
 if (cross > 0) {  
 isPositive = true;  
 } else if (cross < 0) {  
 isNegative = true;  
 }  
 }  
  
 if (isPositive && isNegative) {  
 isConvex = false;  
 } else {  
 isConvex = true;  
 }  
  
 if (isConvex) {  
 System.out.println("Многоугольник выпуклый.");  
 } else {  
 System.out.println("Многоугольник не выпуклый.");  
 }  
  
 }  
}

**Скриншоты:**

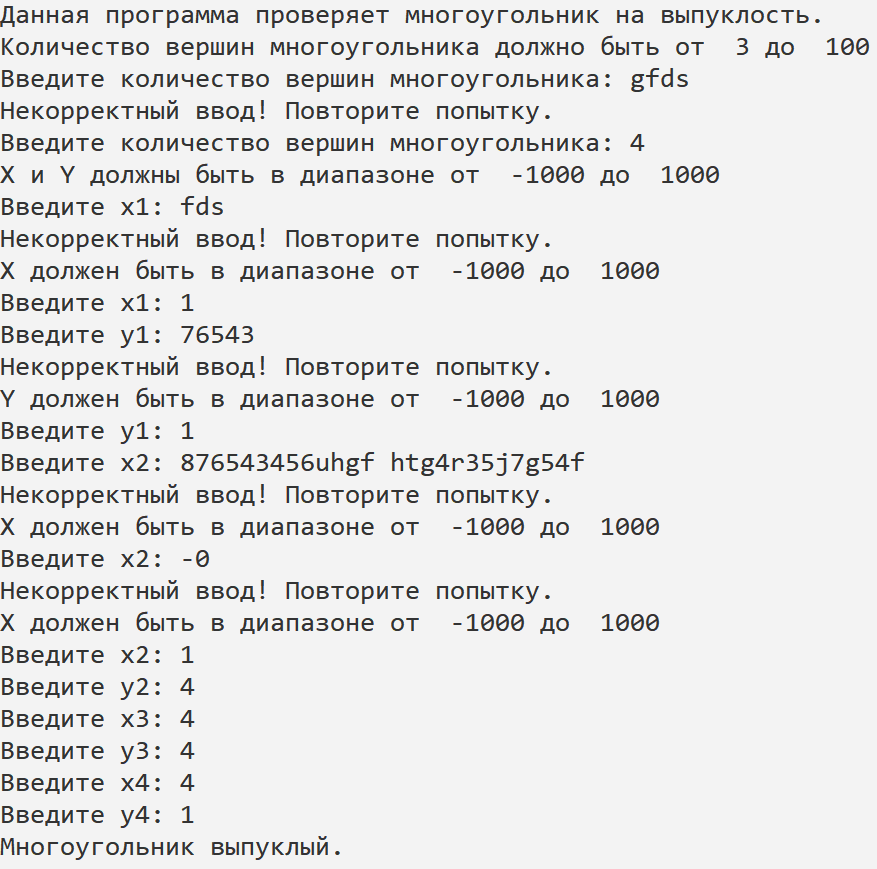
**Delphi:**

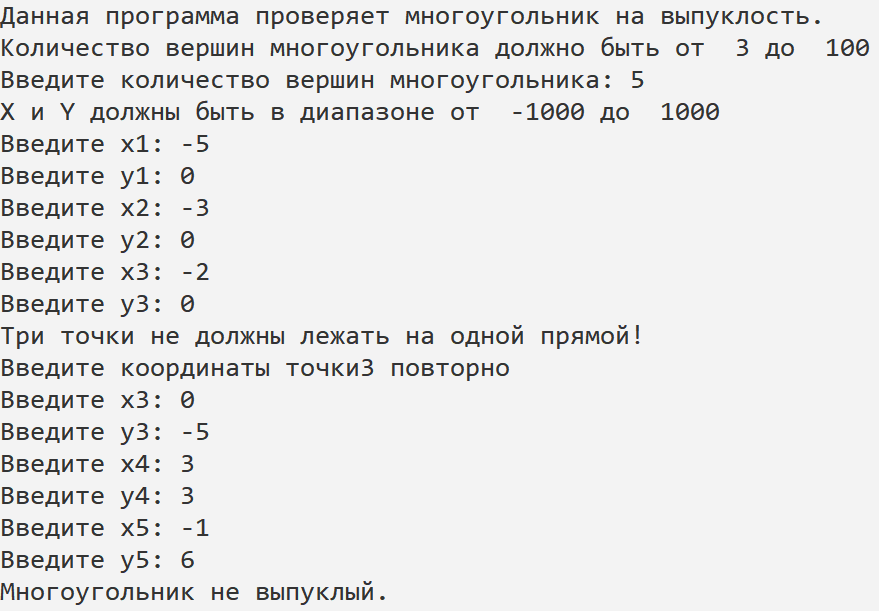
****

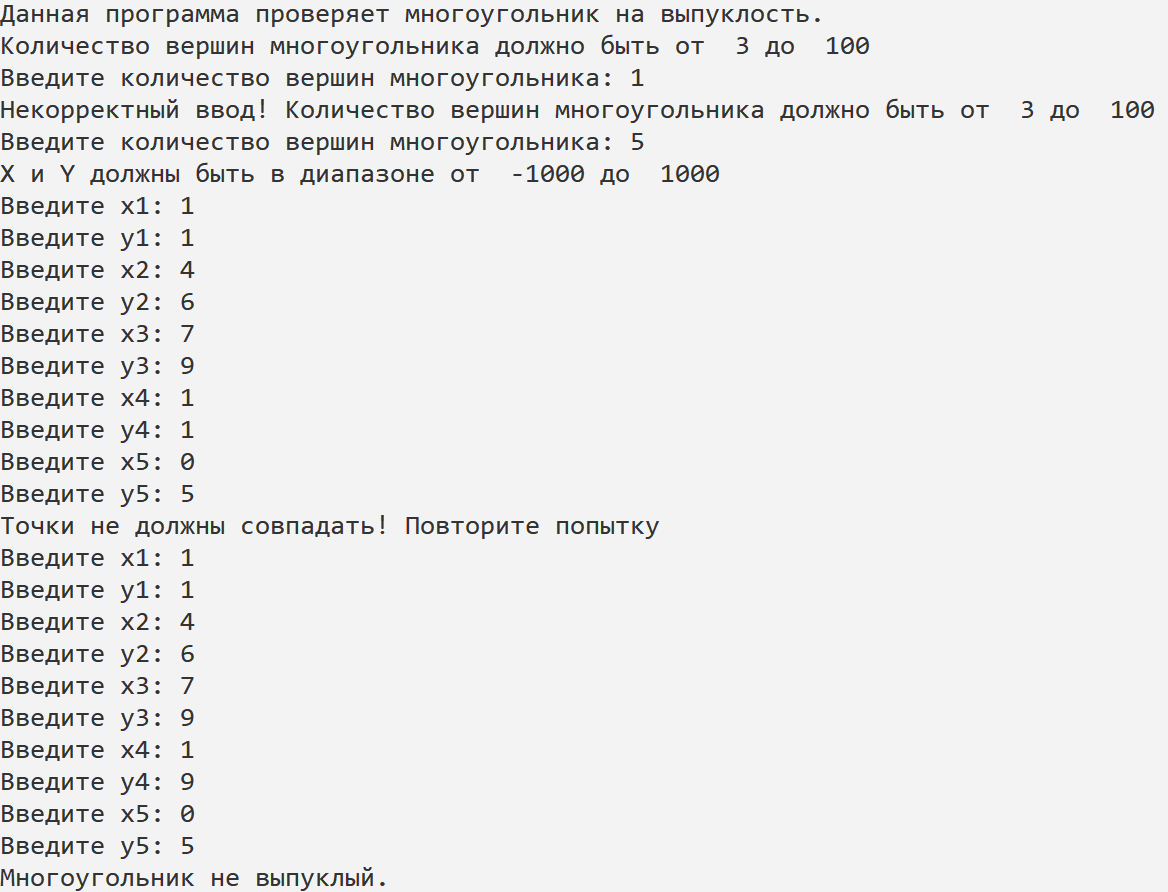
****

****

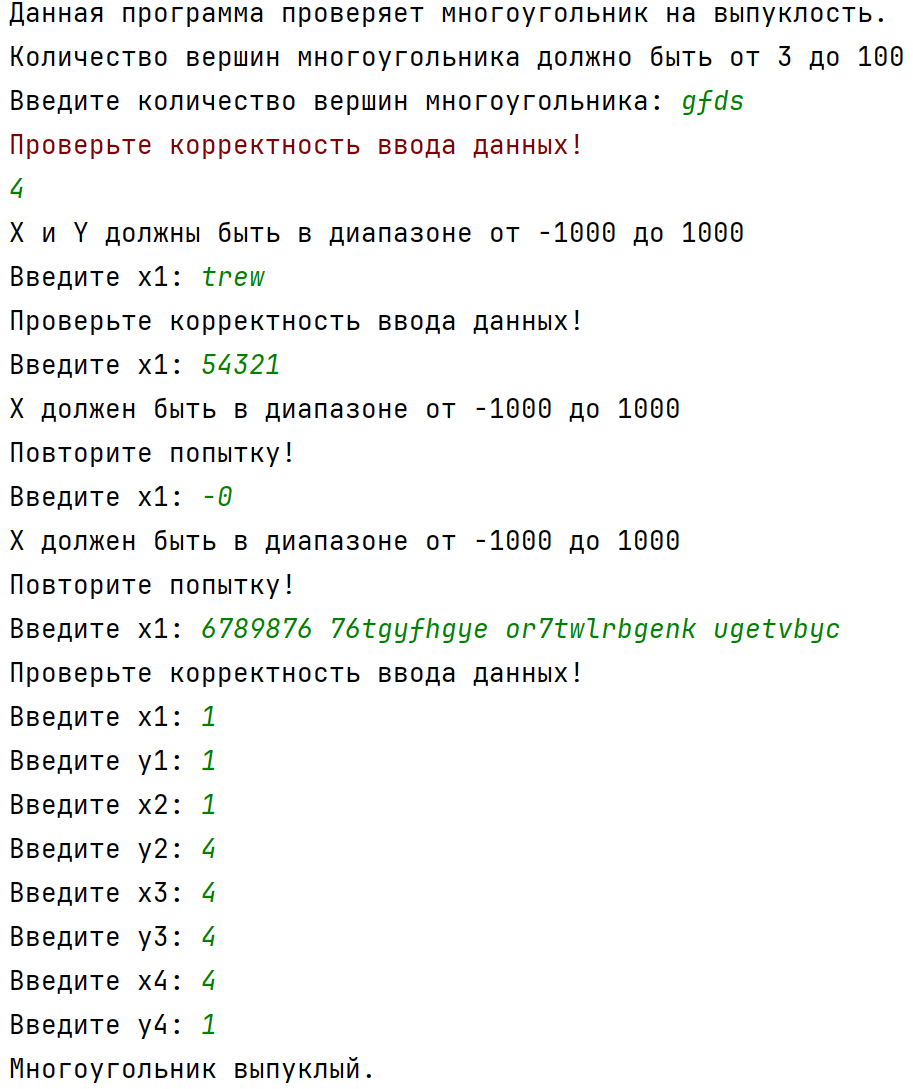
**C++:**

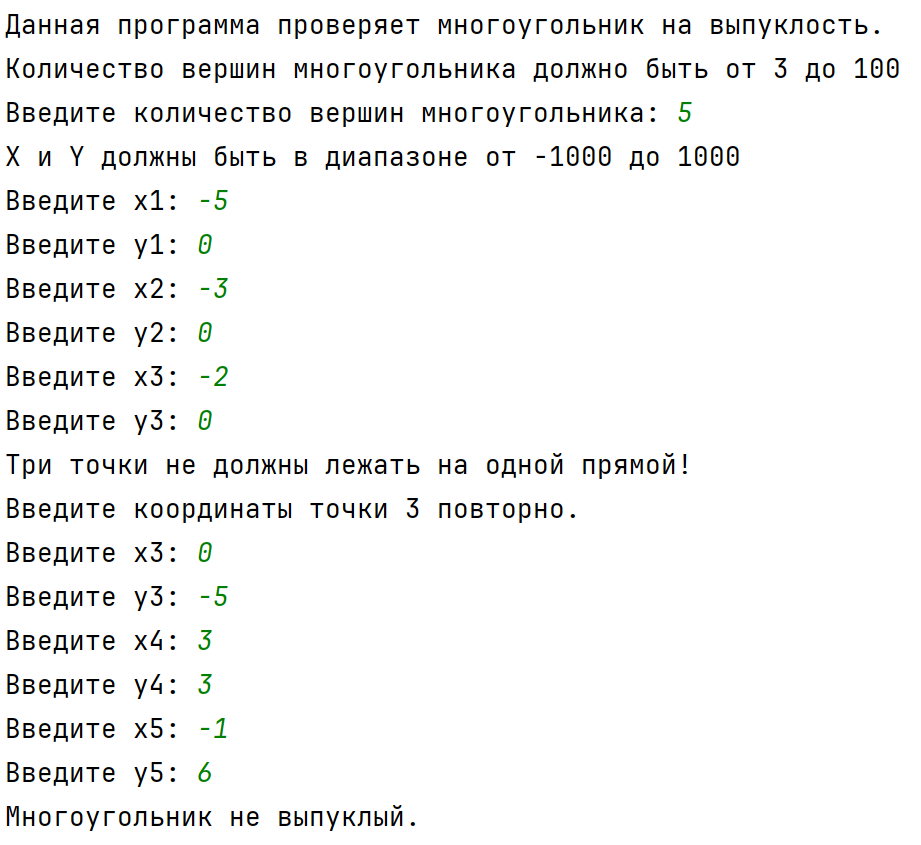
****

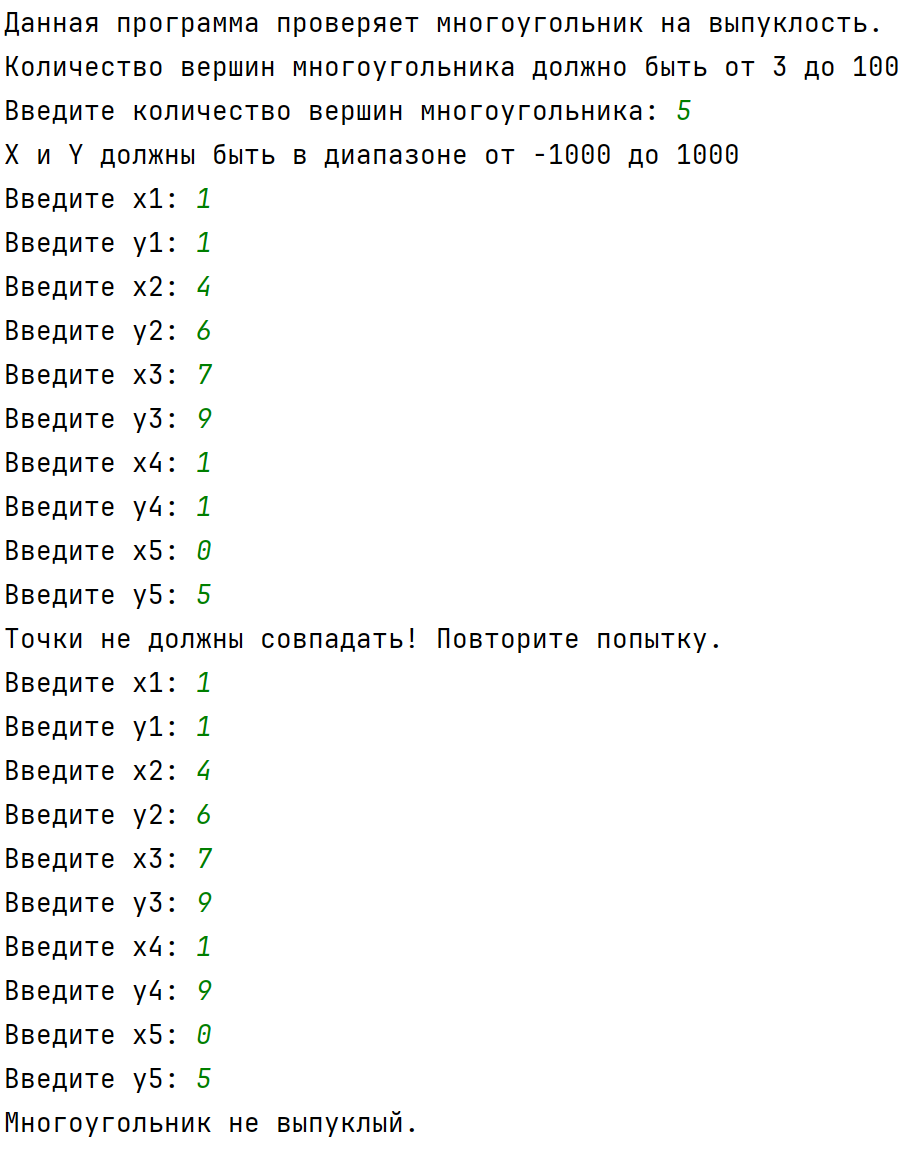
****

****

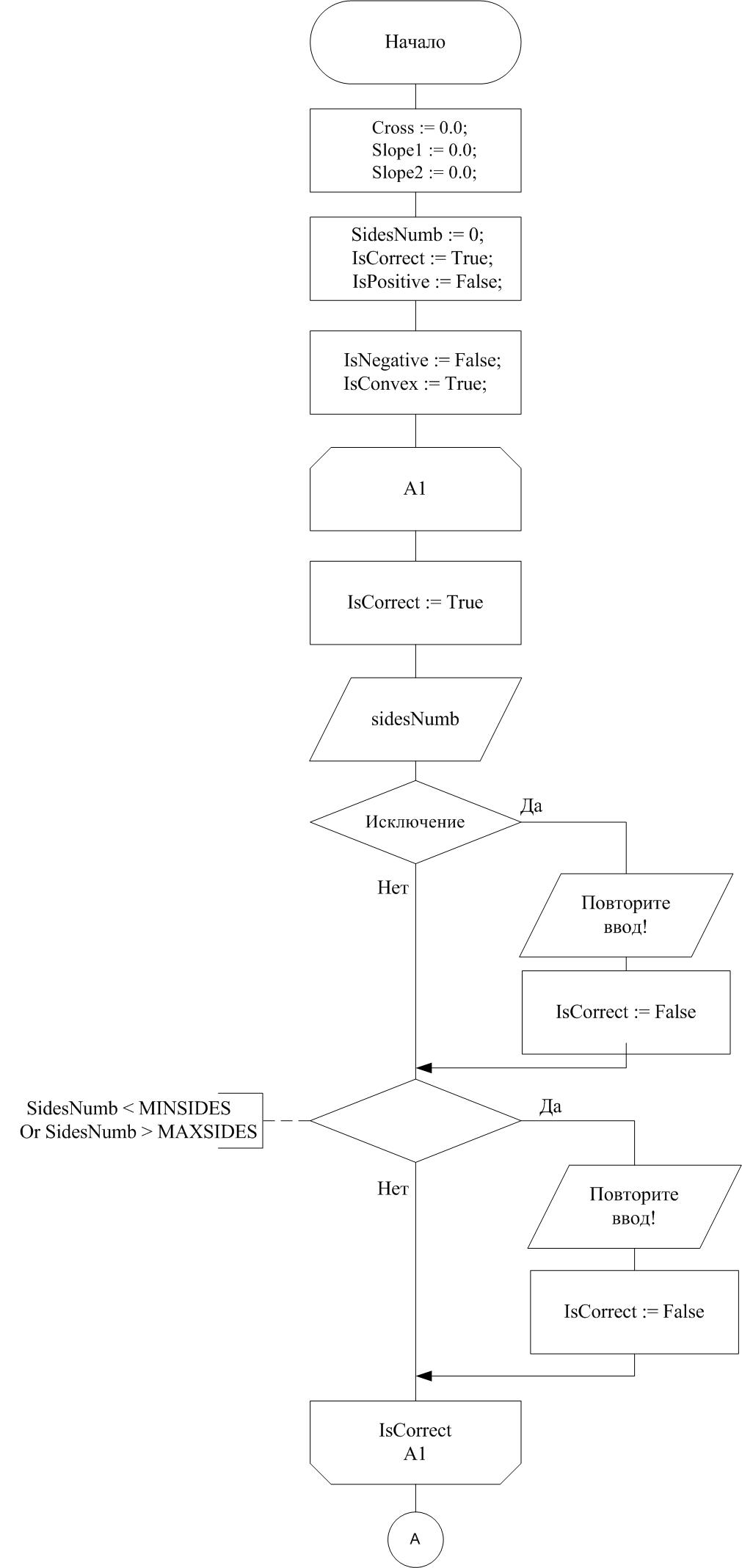
**Java:**

****

****

****

**Блок-схема:**

****

