УО «Белорусский государственный университет информатики и радиоэлектроники»

Кафедра ПОИТ

Отчет по лабораторной работе №2.3

по предмету

Основы алгоритмизации и программирования

Вариант 14

Выполнил:

Крутько А.А.

Проверила:

Данилова Г.В.

Группа 251004

Минск 2022

Задание:

3. Найти «седловую точку» заданной матрицы.

Код программы на **Delphi**:

Program LabThirdBlockSecond;

Uses

System.SysUtils;

Type

TArrArrOI = Array of Array of Integer;

Function ConsoleInputChoice(): Boolean;

Var

IsIncorrect: Boolean;

IsTrue: Boolean;

K: Byte;

Const

MIN\_NUM = 0;

MAX\_NUM = 1;

Begin

Repeat

IsIncorrect := True;

Try

Readln(K);

Except

Write('Повторите ввод: ');

IsIncorrect := False;

End;

If IsIncorrect and ((K > MAX\_NUM) or (K < MIN\_NUM)) Then

Begin

Write('Повторите ввод: ');

IsIncorrect := False;

End;

Until IsIncorrect;

If K = 0 Then

IsTrue := True

Else

IsTrue := False;

ConsoleInputChoice := IsTrue;

End;

Function ParameterIn(Const MIN\_NUM: Integer): Integer;

Var

IsIncorrect: Boolean;

N: Integer;

Begin

Repeat

IsIncorrect := True;

Write('Введите количество строк: ');

Try

Readln(N);

Except

Writeln('Ошибка, введите целочисленное число');

IsIncorrect := False;

End;

If IsIncorrect and (N < MIN\_NUM) Then

Begin

Write('Введите целочисленное число, большее 1: ');

IsIncorrect := False;

End;

Until IsIncorrect;

ParameterIn := N;

End;

Function ParameterInF(Var F: TextFile): Integer;

Var

N: Integer;

Begin

Readln(F, N);

ParameterInF := N;

End;

Function FillMatrix(A: TArrArrOI): TArrArrOI;

Var

I, J: Integer;

IsIncorrect: Boolean;

Begin

For I := 0 to High(A) do

begin

For J := 0 to High(A[0]) do

Begin

Repeat

Write('Введите ячейку ', I + 1, ' строки ', J + 1, ' столбца: ');

IsIncorrect := True;

Try

Readln(A[I,J]);

Except

Writeln('Ошибка, введите целочисленное число');

IsIncorrect := False;

End;

Until IsIncorrect;

End;

End;

FillMatrix := A;

End;

Function FillMatrixF(Var F: TextFile; A: TArrArrOI): TArrArrOI;

Var

I, J: Integer;

Begin

For I := 0 to High(A) do

Begin

For J := 0 to High(A[0]) do

Read(F, A[I,J]);

End;

FillMatrixF := A;

End;

Procedure WriteMatrix(A: TArrArrOI);

Var

I, J: Integer;

Begin

For I := 0 to High(A) do

Begin

For J := 0 to High(A[0]) do

Write(Format('%7d', [A[I,J]]));

Writeln;

End;

End;

Procedure WriteMatrixF(Var Out: TextFile; A: TArrArrOI);

Var

I, J: Integer;

Begin

For I := 0 to High(A) do

begin

For J := 0 to High(A[0]) do

Write(Out, Format('%7d', [A[I,J]]));

Writeln(Out);

End;

End;

Function MinStroka(A: TArrArrOI; I: Integer): Integer;

Var

J, StrokaMin: Integer;

Begin

StrokaMin := A[I,0];

For J := 1 to High(A[0]) Do

Begin

If A[I,J] < StrokaMin Then

StrokaMin := A[I,J];

End;

MinStroka := StrokaMin;

End;

Function MaxStolbets(A: TArrArrOI; StrokaMin: Integer; I: Integer): Integer;

Var

J, StolbetsMax: Integer;

Begin

For J := 0 to High(A[0]) Do

Begin

If A[I,J] = StrokaMin Then

StolbetsMax := J;

End;

MaxStolbets := StolbetsMax;

End;

Function SedlovayaTochka(A: TArrArrOI; StolbetsMax: Integer; StrokaMin: Integer; I: Integer): Boolean;

Var

M: Integer;

IsTrue: Boolean;

Begin

M := 0;

IsTrue := True;

While (M <= High(A)) and (IsTrue) do

Begin

If StrokaMin < A[M, StolbetsMax] Then

IsTrue := False

Else

Inc(M);

End;

SedlovayaTochka := IsTrue;

End;

Procedure WriteSedlovaya(IsTrue: Boolean; StolbetsMax: Integer; StrokaMin: Integer; I: Integer);

Begin

If IsTrue Then

Writeln('Седловая точка в ', I + 1, ' строке и ', StolbetsMax + 1, ' столбце   
 = ', StrokaMin);

End;

Procedure WriteSedlovayaF(Var FileOut: TextFile; IsTrue: Boolean; StolbetsMax: Integer; StrokaMin: Integer; I: Integer);

Begin

If IsTrue Then

Writeln(FileOut, 'Седловая точка в ', I + 1, ' строке и ', StolbetsMax + 1, '   
 столбце = ', StrokaMin)

End;

Function WayToFile(P: Integer): String;

Var

Way: String;

F: Text;

IsIncorrect: Boolean;

Arr: TArrArrOI;

Cell, N, L, I, J, Index1, Index2: Integer;

Begin

Repeat

IsIncorrect := true;

Writeln('Введите путь к файлу');

Readln(Way);

AssignFile(F, Way);

If FileExists(Way) Then

Begin

If P = 1 Then

Begin

Reset(F);

Try

Read(F, N);

Except

Writeln('Проверьте правильность ввода вашей строки в файле на   
 наличие целого числа');

IsIncorrect := False;

End;

If N < 2 Then

Begin

Writeln('Ошибка, ваше целочисленное число для строк   
 должно быть больше 1');

IsIncorrect := False;

End;

Try

Read(F, L);

Except

Writeln('Проверьте правильность ввода вашего столбца в файле   
 на наличие целого числа');

IsIncorrect := False;

End;

If L < 2 Then

Begin

Writeln('Ошибка, ваше целочисленное число для столбца   
 должно быть больше 1');

IsIncorrect := False;

End;

SetLength(Arr, N, L);

For I := 0 to High(Arr) Do

Begin

For J := 0 to High(Arr[0]) do

Begin

Try

Read(F, Cell);

Except

IsIncorrect := False;

Index2 := J;

Index1 := I;

Writeln('Ошибка, проверьте параметр документа ввода матрицы ', Index1+1,' строки ', Index2+1, ' столбца на соответствие целого числа');

End;

End;

End;

Close(F);

End;

End

Else

Begin

Writeln('Файла в заданном пути нет');

IsIncorrect := False;

End;

Until IsIncorrect;

WayToFile := Way;

End;

Function MatrixIn(IsInput: Boolean; Const MIN\_NUM: Integer): TArrArrOI;

Var

N, L: Integer;

A: TArrArrOI;

Way: String;

F: TextFile;

Begin

If IsInput Then

Begin

N := ParameterIn(MIN\_NUM);

L := ParameterIn(MIN\_NUM);

SetLength(A, N, L);

A := FillMatrix(A);

End

Else

Begin

Way := WayToFile(1);

AssignFile(F, Way);

Reset(F);

N := ParameterInF(F);

L := ParameterInF(F);

SetLength(A, N, L);

A := FillMatrixF(F, A);

Close(F);

Writeln('С файла все считано успешно');

End;

MatrixIn := A;

End;

Procedure MatrixOut(isOutput: Boolean; A: TArrArrOI);

Var

I, J, StrokaMin, StolbetsMax: Integer;

IsTrue: Boolean;

Way: String;

FileOut: TextFile;

Begin

If IsOutput Then

Begin

WriteMatrix(A);

For I := 0 to High(A) do

Begin

StrokaMin := MinStroka(A, I);

StolbetsMax := MaxStolbets(A, StrokaMin, I);

IsTrue := SedlovayaTochka(A, StolbetsMax, StrokaMin, I);

WriteSedlovaya(IsTrue, StolbetsMax, StrokaMin, I);

End;

If Not(IsTrue) Then

Writeln('Седловых точек нет');

End

Else

Begin

Way := WayToFile(11);

Assign(FileOut, Way);

ReWrite(FileOut);

WriteMatrixF(FileOut, A);

For I := 0 to High(A) do

Begin

StrokaMin := MinStroka(A, I);

StolbetsMax := MaxStolbets(A, StrokaMin, I);

IsTrue := SedlovayaTochka(A, StolbetsMax, StrokaMin, I);

WriteSedlovayaF(FileOut, IsTrue, StolbetsMax, StrokaMin, I);

End;

If Not(IsTrue) Then

Writeln(FileOut, 'Седловых точек нет');

Close(FileOut);

End;

End;

Var

A: TArrArrOI;

IsInput, IsOutput: Boolean;

Const

MIN\_NUM = 2;

Begin

Writeln('Данная программа находит седловую точку в матрице');

Write('Введите 0, если хотите ввод на консоль, 1 — ввод из файла: ');

IsInput := ConsoleInputChoice();

A := MatrixIn(IsInput, MIN\_NUM);

Write('Введите 0, если хотите вывод на консоль, 1 — вывод в файл: ');

IsOutput := ConsoleInputChoice();

MatrixOut(IsOutput, A);

Readln;

End.

Код программы на **C++**:

#include <iostream>

#include <fstream>

#include <string>

#include <iomanip>

using namespace std;

bool consoleChoice()

{

bool isIncorrect, isTrue;

int k;

const int MIN\_NUM = 0, MAX\_NUM = 1;

do {

isIncorrect = false;

cin >> k;

if (cin.fail())

{

cin.clear();

while (cin.get() != '\n');

cout << "Проверьте правильность ввода данных: " << endl;

isIncorrect = true;

}

if (!isIncorrect && cin.get() != '\n')

{

cin.clear();

while (cin.get() != '\n');

cout << "Проверьте правильность ввода данных: " << endl;

isIncorrect = true;

}

if (!isIncorrect && (k < MIN\_NUM || k > MAX\_NUM)) {

cout << "Ваше число долнно быть или 1, или 0: " << endl;

isIncorrect = true;

}

} while (isIncorrect);

if (k == 0)

isTrue = true;

else

isTrue = false;

return isTrue;

}

int parameterIn(const int MIN\_NUM) {

bool isIncorrect;

int n;

do {

isIncorrect = false;

cin >> n;

if (cin.fail())

{

cin.clear();

while (cin.get() != '\n');

cout << "Проверьте правильность ввода данных: " << endl;

isIncorrect = true;

}

if (!isIncorrect && cin.get() != '\n')

{

cin.clear();

while (cin.get() != '\n');

cout << "Проверьте правильность ввода данных: " << endl;

isIncorrect = true;

}

if (!isIncorrect && n < MIN\_NUM) {

cout << "Должно быть > 1" << endl;

isIncorrect = true;

}

} while (isIncorrect);

return n;

}

int takeCell()

{

bool isIncorrect;

int n;

do {

isIncorrect = false;

cin >> n;

if (cin.fail())

{

cin.clear();

while (cin.get() != '\n');

cout << "Проверьте правильность ввода данных: " << endl;

isIncorrect = true;

}

if (!isIncorrect && cin.get() != '\n')

{

cin.clear();

while (cin.get() != '\n');

cout << "Проверьте правильность ввода данных: " << endl;

isIncorrect = true;

}

} while (isIncorrect);

return n;

}

int\*\* fillMatrix(int\*\* arr, int nRow, int nCol)

{

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

{

for (int j = 0; j < nCol; j++)

{

cout << "Элемент " << (i+1) << " строки " << (j+1) << " столбца = ";

arr[i][j] = takeCell();

}

}

return arr;

}

void writeMatrix(int\*\* arr, int nRow, int nCol)

{

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

{

for (int j = 0; j < nCol; j++)

{

cout << setw(5) << arr[i][j];

}

cout << endl;

}

}

int minRow(int\*\* arr, int i, int nCol)

{

int rowMin;

rowMin = arr[i][0];

for (int j = 1; j < nCol; j++)

{

if (arr[i][j] < rowMin)

rowMin = arr[i][j];

}

return rowMin;

}

int maxColumn(int\*\* arr, int rowMin, int i, int nCol)

{

int colMax;

for (int j = 0; j < nCol; j++) {

if (arr[i][j] == rowMin)

{

colMax = j;

}

}

return colMax;

}

bool sedlovayaTochka(int\*\* arr, int rowMin, int colMax, int i, int nRow)

{

bool isTrue;

int m = 0;

isTrue = true;

while (m < nRow && isTrue) {

if (rowMin < arr[m][colMax]) {

isTrue = false;

}

else {

m++;

}

}

return isTrue;

}

void writeSedlovaya(bool isTrue, int colMax, int rowMin, int i)

{

if (isTrue)

cout << "Седловая точка в " << (i + 1) << " строке и " << (colMax + 1) << "   
 столбце = " << rowMin;

}

bool isFileCorrect(string way)

{

int nRow, nCol, size, i = 0;

bool isIncorrect = false;

ifstream fin(way);

fin >> nRow;

if (nRow < 2)

{

isIncorrect = true;

cout << "Число строк в файле должно быть > 2" << endl;

}

fin >> nCol;

if (nCol < 2)

{

isIncorrect = true;

cout << "Число строк в файле должно быть > 2" << endl;

}

size = nRow \* nCol;

while (i < size && !isIncorrect)

{

fin >> i;

i++;

}

fin.close();

return isIncorrect;

}

string wayToFile(int n)

{

string way;

bool isIncorrect;

do

{

isIncorrect = false;

cout << "Введите путь к файлу" << endl;

cin >> way;

if (n == 1)

{

ifstream fin(way);

if (!fin.is\_open())

{

cout << "Проверьте правильность и нахождение вашего файла по   
 заданному пути" << endl;

isIncorrect = true;

}

else

{

isIncorrect = isFileCorrect(way);

}

fin.close();

}

else

{

ifstream fout(way);

if (!fout.is\_open())

{

cout << "Проверьте правильность и нахождение вашего файла по   
 заданному пути" << endl;

isIncorrect = true;

}

fout.close();

}

} while (isIncorrect);

return way;

}

int\*\* createMatrix(int nRow, int nCol)

{

int\*\* arr;

arr = new int\* [nRow];

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

{

arr[i] = new int[nCol];

}

return arr;

}

int\*\* takeArrFromFile(string way, int\*\* arr, int nRow, int nCol)

{

ifstream fin(way);

fin.seekg(4, ios::beg);

arr = createMatrix(nRow, nCol);

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

{

for (int j = 0; j < nCol; j++)

fin >> arr[i][j];

}

fin.close();

return arr;

}

void writeMatrixF(string way, int\*\* arr, int nRow, int nCol)

{

ofstream fout(way);

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

{

for (int j = 0; j < nCol; j++)

{

fout << setw(5) << arr[i][j];

}

fout << endl;

}

fout.close();

}

void writeSedlovayaF(string way, bool isTrue, int colMax, int rowMin, int i)

{

ofstream fout(way, ios::app); //открывает файл для дозаписи, старые данные не стирает

if (isTrue)

fout << "Седловая точка в " << (i + 1) << " строке и " << (colMax + 1) << "   
 столбце = " << rowMin;

fout.close();

}

int parameterInF(string way, int p)

{

int n;

ifstream fin(way);

if (p == 1)

{

fin.seekg(0, ios::beg);

fin >> n;

}

else

{

fin.seekg(2, ios::beg);

fin >> n;

}

fin.close();

return n;

}

int\*\* finalMatrixIn(const int MIN\_NUM, bool isInput, int &nRow, int &nCol)

{

int\*\* arr = nullptr;

string way;

if (isInput)

{

cout << "Введите количество строк: ";

nRow = parameterIn(MIN\_NUM);

cout << "Введите количество столбцов: ";

nCol = parameterIn(MIN\_NUM);

arr = createMatrix(nRow, nCol);

arr = fillMatrix(arr, nRow, nCol);

}

else

{

way = wayToFile(1);

nRow = parameterInF(way, 1);

nCol = parameterInF(way, 0);

arr = takeArrFromFile(way, arr, nRow, nCol);

}

return arr;

}

void finalMatrixOutput(bool isOutput, int\*\* arr, int nRow, int nCol)

{

int rowMin, colMax;

bool isTrue;

string way;

if (isOutput)

{

writeMatrix(arr, nRow, nCol);

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

{

rowMin = minRow(arr, i, nCol);

colMax = maxColumn(arr, rowMin, i, nCol);

isTrue = sedlovayaTochka(arr, rowMin, colMax, i, nRow);

writeSedlovaya(isTrue, colMax, rowMin, i);

}

if (!(isTrue))

{

cout << "Седловых точек нет";

}

}

else

{

way = wayToFile(11);

writeMatrixF(way, arr, nRow, nCol);

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

{

rowMin = minRow(arr, i, nCol);

colMax = maxColumn(arr, rowMin, i, nCol);

isTrue = sedlovayaTochka(arr, rowMin, colMax, i, nRow);

writeSedlovayaF(way, isTrue, colMax, rowMin, i);

}

ofstream fout(way, ios::app);

if (!(isTrue))

{

fout << "Седловых точек нет";

}

fout.close();

}

}

void clearMemory(int\*\* arr, int nRow)

{

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

{

delete[] arr[i];

}

delete[] arr;

}

int main()

{

setlocale(LC\_ALL, "Russian");

const int MIN\_NUM = 2;

bool isInput, isOutput;

int nRow = NULL, nCol = NULL;

int\*\* arr = nullptr;

cout << "Данная программа находит седловую точку в матрице" << endl;

cout << "Введите 0, если хотите ввод на консоль, 1 — ввод из файла: ";

isInput = consoleChoice();

arr = finalMatrixIn(MIN\_NUM, isInput, nRow, nCol);

cout << "Введите 0, если хотите ввод на консоль, 1 — ввод из файла: ";

isOutput = consoleChoice();

finalMatrixOutput(isOutput, arr, nRow, nCol);

clearMemory(arr, nRow);

return 0;

}

Код программы на **Java**:

import java.io.\*;  
import java.util.Scanner;  
public class LabThirdBlockSecond {  
 static Scanner scan = new Scanner(System.in);  
 public static boolean consoleChoice(){  
 boolean isIncorrect, isTrue;  
 byte k = 0;  
 final int MIN\_NUM = 0, MAX\_NUM = 1;  
 do {  
 isIncorrect = false;  
 try {  
 k = Byte.parseByte(scan.nextLine());  
 } catch (Exception E) {  
 System.err.println("Повторите ввод: ");  
 isIncorrect = true;  
 }  
 if (!isIncorrect && (k < MIN\_NUM || k > MAX\_NUM)){  
 System.err.println("Повторите ввод: ");  
 isIncorrect = true;  
 }  
 } while (isIncorrect);  
 if (k == 0)  
 isTrue = true;  
 else  
 isTrue = false;  
 return isTrue;  
 }  
 public static int strokaIn(final int MIN\_NUM){  
 boolean isIncorrect;  
 int n = 0;  
 do {  
 System.out.println("Введите количество строк: ");  
 isIncorrect = false;  
 try {  
 n = Integer.parseInt(scan.nextLine());  
 } catch (Exception E) {  
 System.err.println("Повторите ввод: ");  
 isIncorrect = true;  
 }  
 if (!isIncorrect && (n < MIN\_NUM)){  
 System.err.println("Повторите ввод: ");  
 isIncorrect = true;  
 }  
 } while (isIncorrect);  
 return n;  
 }  
 public static int stolbetsIn(final int MIN\_NUM){  
 boolean isIncorrect;  
 int l = 0;  
 do {  
 System.out.println("Введите количество столбцов: ");  
 isIncorrect = false;  
 try {  
 l = Integer.parseInt(scan.nextLine());  
 } catch (Exception E) {  
 System.err.println("Повторите ввод: ");  
 isIncorrect = true;  
 }  
 if (!isIncorrect && (l < MIN\_NUM)){  
 System.err.println("Повторите ввод: ");  
 isIncorrect = true;  
 }  
 } while (isIncorrect);  
 return l;  
 }  
 public static int[][] fillMatrix(int[][] arr) {  
 boolean isIncorrect;  
 for (int i = 0; i < arr.length; i++) {  
 for (int j = 0; j < arr[0].length; j++) {  
 do {  
 System.out.println("Введите ячейку "+ (i+1) + " строки "+   
 (j+1) + " столбца: ");  
 isIncorrect = false;  
 try {  
 arr[i][j] = Integer.parseInt(scan.nextLine());  
 } catch (Exception E) {  
 System.err.println("Ошибка, введите целочисленное   
 число");  
 isIncorrect = true;  
 }  
 } while(isIncorrect);  
 }  
 }  
 return arr;  
 }  
 public static void writeMatrix(int [][] arr) {  
 for (int i = 0; i < arr.length; i++){  
 for (int j = 0; j < arr[0].length; j++)  
 System.out.printf("%5d", arr[i][j]);  
 System.out.print("\n");  
 }  
 }  
 public static int minStroka(int[][] arr, int i){  
 int strokaMin;  
 strokaMin = arr[i][0];  
 for (int j = 1; j < arr[0].length; j++) {  
 if (arr[i][j] < strokaMin)  
 strokaMin = arr[i][j];  
 }  
 return strokaMin;  
 }  
 public static int maxStolbets(int [][]arr, int strokaMin, int i){  
 int stolbetsMax = 0;  
 for (int j = 0; j < arr[0].length; j++) {  
 if (arr[i][j] == strokaMin)  
 stolbetsMax = j;  
 }  
 return stolbetsMax;  
 }  
 public static boolean sedlovayaTochka(int [][] arr, int strokaMin, int stolbetsMax, int i) {  
 boolean isTrue;  
 int m = 0;  
 isTrue = true;  
 while (m < arr.length && isTrue) {  
 if (strokaMin < arr[m][stolbetsMax]) {  
 isTrue = false;  
 } else {  
 m++;  
 }  
 }  
 return isTrue;  
 }  
  
 public static void writeSedlovaya(boolean isTrue, int stolbetsMax, int strokaMin,   
 int i) {  
 if (isTrue)  
 System.out.println("Седловая точка в " + (i + 1) + " строке и " +   
 (stolbetsMax + 1) + " столбце = " + strokaMin);  
 }  
  
 public static String takeFileWay(int x) {  
 String way;  
 boolean isIncorrect;  
 do {

isIncorrect = false;  
 System.out.println("Введите путь к файлу");  
 way = scan.nextLine();  
 File file = new File(way);  
 if (!file.exists() || !file.canWrite() ||!file.canRead() || !way.endsWith(".txt") || file.isDirectory()) {  
 isIncorrect = true;  
 System.err.println("Файл не найден, или неверный формат файла; повторите попытку ввода");  
 } else if (x == 1) {  
 isIncorrect = isTakeInformationFromFile(way);  
 }  
 } while (isIncorrect);  
 return way;  
 }  
  
 public static boolean isTakeInformationFromFile(String way) {  
 int n = 0, l = 0;  
 boolean isIncorrect = false;  
 try (BufferedReader br = new BufferedReader(new FileReader(way))) {  
 n = Integer.parseInt(br.readLine());  
 if (n < 2) throw new Exception ("Количество строк должно быть >   
 1");  
 l = Integer.parseInt(br.readLine());  
 if (l < 2) throw new Exception ("Количество столбцов должно быть   
 > 1");  
 for (int i = 0; i < n\*l; i++) {

try {  
 Integer.parseInt(br.readLine());

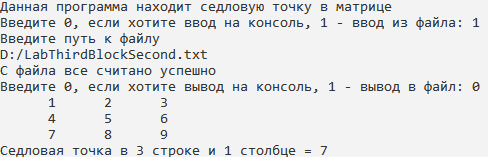
} catch (Exception E) {

isIncorrect = true;

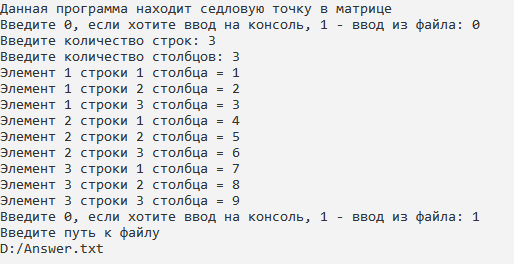
System.out.println("Проверьте правильность элементов вашей матрицы");

}  
 }  
 if (br.readLine() != null)  
 isIncorrect = true;  
 } catch (Exception e) {  
 System.err.println("Проверьте правильность данных матрицы,   
 введенной в документе");  
 System.out.println(e.getMessage());  
 isIncorrect = true;  
 } finally {  
 if (isIncorrect)  
 System.err.println("Уберите лишние данные");  
 }  
 return isIncorrect;  
 }  
  
 public static int[][] takeArrFromFile(String way){  
 int[][] arr = new int[0][0];  
 int n = 0, l = 0;  
 try (BufferedReader br = new BufferedReader(new FileReader(way))) {  
 n = Integer.parseInt(br.readLine());  
 l = Integer.parseInt(br.readLine());  
 arr = new int[n][l];  
 for (int i = 0; i < arr.length; i++) {  
 for (int j = 0; j < arr[0].length; j++) {  
 arr[i][j] = Integer.parseInt(br.readLine());  
 }  
 }  
 } catch (Exception e) {  
 System.err.println("Проверьте правильность данных, введенных в документе");  
 }  
 return arr;  
 }  
  
 public static void writeMatrixF(String way, int[][] arr){  
 try (PrintWriter pw = new PrintWriter(new FileWriter(way, false))) { for (int i = 0; i < arr.length; i++) { //(false тут, чтобы с прошлой программы перезаписывать то, что было раньше и начинать с 0  
 for (int j = 0; j < arr[0].length; j++) {  
 pw.printf("%3d", arr[i][j]);  
 }  
 pw.write("\n");  
 }  
 } catch (Exception e) {  
 System.err.println("Ошибка файла");  
 System.out.println(e.getMessage());  
 }  
 }  
  
 public static void writeSedlovayaF(String way, boolean isTrue, int   
 stolbetsMax, int strokaMin, int i) {  
 try (FileWriter fileOut = new FileWriter(way, true)) {  
 if (isTrue)  
 fileOut.write("Седловая точка в " + (i + 1) + " строке и " +   
 (stolbetsMax + 1) + " столбце = " + strokaMin);  
 } catch (Exception e) {  
 System.err.println("Ошибка файла");  
 System.out.println(e.getMessage());  
 }  
 }  
  
 public static int[][] matrixIn(boolean isInput, final int MIN\_NUM) {  
 int n, l;  
 String way;  
 int [][] arr;  
 if (isInput) {  
 n = strokaIn(MIN\_NUM);  
 l = stolbetsIn(MIN\_NUM);  
 arr = new int[n][l];  
 arr = fillMatrix(arr);  
 } else {  
 way = takeFileWay(1);  
 arr = takeArrFromFile(way);  
 }  
 return arr;  
 }  
  
 public static void matrixOut(int [][] arr, boolean isOutput) {  
 int strokaMin, stolbetsMax;  
 boolean isTrue = false;  
 String way;  
 if (isOutput) {  
 writeMatrix(arr);  
 for (int i = 0; i < arr.length; i++) {  
 strokaMin = minStroka(arr, i);  
 stolbetsMax = maxStolbets(arr, strokaMin, i);  
 isTrue = sedlovayaTochka(arr, strokaMin, stolbetsMax, i);  
 writeSedlovaya(isTrue, stolbetsMax, strokaMin, i);  
 }  
 if (!(isTrue)) {  
 System.out.println("Седловых точек нет");  
 }  
 } else {  
 way = takeFileWay(11);  
 writeMatrixF(way, arr);  
 for (int i = 0; i < arr.length; i++) {  
 strokaMin = minStroka(arr, i);  
 stolbetsMax = maxStolbets(arr, strokaMin, i);  
 isTrue = sedlovayaTochka(arr, strokaMin, stolbetsMax, i);  
 writeSedlovayaF(way, isTrue, stolbetsMax, strokaMin, i);  
 }  
 try (FileWriter fileOut = new FileWriter(way, true)) {  
 if (!(isTrue)) {  
 fileOut.write("\nСедловых точек нет");  
 }  
 } catch(Exception e) {  
 System.err.println("Ошибка файла");  
 System.out.println(e.getMessage());  
 }  
 }  
 }  
  
 public static void main(String[] args) {  
 boolean isInput, isOutput;  
 String way;  
 int arr[][];  
 final int MIN\_NUM = 2;  
 System.out.println("Данная программа находит седловую точку в   
 матрице");  
 System.out.println("Введите 0, если хотите ввод на консоль, 1 — ввод   
 из файла: ");  
 isInput = consoleChoice();  
 arr = matrixIn(isInput, MIN\_NUM);  
 System.out.println("Введите 0, если хотите вывод на консоль, 1 –   
 вывод в файл: ");  
 isOutput = consoleChoice();  
 matrixOut(arr, isOutput);  
 scan.close();  
 }  
  
 }

Результат на **Delphi**:



Результат на **C++**:

****

Результат на **Java**:

