УО «Белорусский государственный университет информатики и радиоэлектроники»

Кафедра ПОИТ

Отчет по лабораторной работе №2.4

по предмету

Основы алгоритмизации и программирования

Вариант 14

Выполнил:

Крутько А.А.

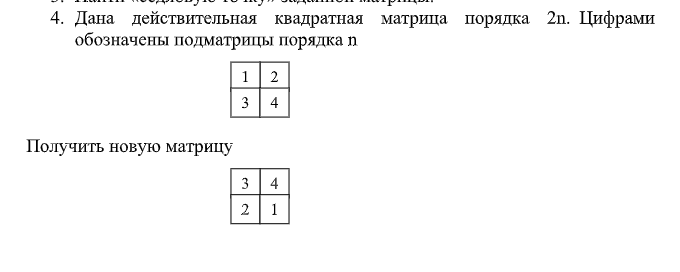
Проверила:

Данилова Г.В.

Группа 251004

Минск 2022

Задание:

Код программы на **Delphi**:

Program LabFourthBlockSecond;

Uses

System.sysUtils;

Type

TArrArrOI = Array of Array of Integer;

Function IsConsoleChoice(): 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;

IsConsoleChoice := IsTrue;

End;

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

Var

Numb:Integer;

isIncorrect: Boolean;

Begin

Repeat

Try

isIncorrect := True;

Readln(Numb);

Except

Writeln('Проверьте правильность введенных данных');

isIncorrect := False;

End;

If (IsIncorrect and (Numb < MIN\_NUM)) Then

Begin

isIncorrect := False;

Writeln('Ваш размер N должен быть больше 1: ');

End;

Until isIncorrect;

DimensionIn := Numb;

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

Write('Ячейка ', I + 1,' строки ', J + 1, ' столбца = ');

Repeat

Try

Readln(A[I,J]);

IsIncorrect := True;

Except

IsIncorrect := False;

Writeln('Повторите ввод');

End;

Until IsIncorrect;

End;

End;

FillMatrix := A;

End;

Function DimensionF(Var F: TextFile): Integer;

Var

N: Integer;

Begin

Readln(F,N);

DimensionF := N;

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 WriteFirstMatrix(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;

Function ConvertMatrix(A: TArrArrOI; N: Integer): TArrArrOI;

Var

I, J, L, X: Integer;

Begin

L := N - 1;

For I := 0 to L do

For J := 0 to L do

Begin

X := A[I,J];

A[I,J]:= A[I + N,J];

A[I + N,J] := A[I,J+N];

A[I,J + N] := A[I + N, J + N];

A[I + N,J + N] := X;

End;

ConvertMatrix := A;

End;

Procedure WriteNewMatrix(A: TArrArrOI; N: Integer);

Var

I, J: Integer;

Begin

A := ConvertMatrix(A, N);

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;

Function WayToFile(P: Integer): String;

Var

Way: String;

F: Text;

IsIncorrect: Boolean;

Cell, N, L, I, J, Index1, Index2, NewN: 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 размерности вашей   
 матрицы');

IsIncorrect := False;

End;

If N < 2 Then

Begin

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

IsIncorrect := False;

End;

NewN := 2\*N;

For I := 1 to NewN Do

Begin

For J := 1 to NewN do

Begin

Try

Read(F, Cell);

Except

IsIncorrect := False;

Index2 := J;

Index1 := I;

Writeln('Ошибка, проверьте параметр документа   
 ввода матрицы ', Index1,' строки ',   
 Index2, ' 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;

Procedure WriteFirstMatrixF(Var FileOut: TextFile; A: TArrArrOI);

Var

I, J: Integer;

Begin

Writeln(FileOut,'Ваша матрица:');

For I := 0 to High(A) Do

Begin

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

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

Writeln(FileOut);

End;

End;

Procedure WriteNewMatrixF(Var FileOut: TextFile; A: TArrArrOI; N: Integer);

Var

I, J: Integer;

Begin

Writeln(FileOut, 'Преобразованная матрица будет выглядеть:');

A := ConvertMatrix(A, N);

For I := 0 to High(A) Do

Begin

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

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

Writeln(FileOut);

End;

End;

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

Var

Way: String;

F: TextFile;

A: TArrArrOI;

Begin

If IsInput Then

Begin

Write('Введите N – размер подматрицы матрицы 2N: ');

N := DimensionIn(MIN\_NUM);

SetLength(A, 2\*N, 2\*N);

A := FillMatrix(A);

End

Else

Begin

Way := WayToFile(1);

AssignFile(F, Way);

Reset(F);

N := DimensionF(F);

SetLength(A, 2\*N, 2\*N);

A := FillMatrixF(F, A);

Close(F);

Writeln('С файла все считано успешно');

End;

MatrixIn := A;

End;

Procedure MatrixOut(IsOutput: Boolean; N: Integer; A: TArrArrOI);

Var

Way: String;

FileOut: TextFile;

Begin

If IsOutput Then

Begin

Writeln('Ваша матрица:');

WriteFirstMatrix(A);

Writeln('Преобразованная матрица будет выглядеть:');

WriteNewMatrix(A, N);

End

Else

Begin

Way := WayToFile(11);

Assign(FileOut, Way);

ReWrite(FileOut);

WriteFirstMatrixF(FileOut, A);

WriteNewMatrixF(FileOut, A, N);

Close(FileOut);

End;

End;

Var

A: TArrArrOI;

N: Integer;

IsInput, IsOutput: Boolean;

Const

MIN\_NUM = 2;

Begin

Writeln('Данная программа меняет подматрицы N вашей матрицы 2N местами');

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

IsInput := IsConsoleChoice();

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

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

IsOutput := IsConsoleChoice();

MatrixOut(IsOutput, N, 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\*\* createMatrix(int n)

{

int\*\* arr;

int doubleN = 2 \* n;

arr = new int\* [doubleN];

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

{

arr[i] = new int[doubleN];

}

return arr;

}

int\*\* fillMatrix(int\*\* arr, int n)

{

int doubleN = 2 \* n;

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

{

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

{

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

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

}

}

return arr;

}

bool isFileCorrect(string way)

{

int n, size, i = 0, elem;

bool isIncorrect = false;

ifstream fin(way);

if (!fin.eof())

{

fin >> n;

if (fin.fail())

{

isIncorrect = true;

cout << "Проверьте правильность введенной размерности матрицы" << endl;

}

else if (!fin.fail() and n < 2)

{

isIncorrect = true;

cout << "Размерность > 1" << endl;

}

}

else

{

isIncorrect = true;

cout << "Нет данных для размерности" << endl;

}

size = 4 \* n \* n;

while (i < size && !isIncorrect)

{

if (!fin.eof())

{

fin >> elem;

i++;

}

else

{

isIncorrect = true;

cout << "Данные при считывании матрицы закончились" << endl;

}

}

if (!fin.eof() && !isIncorrect)

{

isIncorrect = true;

cout << "Уберите лишние данные" << endl;

}

fin.close();

return isIncorrect;

}

string wayToFile(int p)

{

string way;

bool isIncorrect;

do

{

isIncorrect = false;

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

cin >> way;

if (p == 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

{

ofstream 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 parameterInF(string way)

{

int n;

ifstream fin(way);

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

fin >> n;

fin.close();

return n;

}

int\*\* takeArrFromFile(string way, int\*\* arr, int n)

{

int doubleN = 2 \* n;

ifstream fin(way);

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

arr = createMatrix(n);

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

{

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

fin >> arr[i][j];

}

fin.close();

return arr;

}

void writeFirstMatrix(int\*\* arr, int n)

{

int doubleN = 2 \* n;

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

{

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

{

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

}

cout << endl;

}

}

int\*\* convertMatrix(int\*\* arr, int n)

{

int x;

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

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

x = arr[i][j];

arr[i][j] = arr[i + n][j];

arr[i + n][j] = arr[i][j + n];

arr[i][j + n] = arr[i + n][j + n];

arr[i + n][j + n] = x;

}

}

return arr;

}

void writeNewMatrix(int\*\* arr, int n)

{

int doubleN = 2 \* n;

arr = convertMatrix(arr, n);

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

{

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

{

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

}

cout << endl;

}

}

void writeMatrixF(string way, int\*\* arr, int n)

{

int doubleN = 2 \* n;

ofstream fout(way);

fout << "Ваша матрица:" << endl;

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

{

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

{

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

}

fout << endl;

}

fout.close();

}

void writeNewMatrixF(string way, int\*\* arr, int n)

{

int doubleN = 2 \* n;

arr = convertMatrix(arr, n);

ofstream fout(way, ios::app);

fout << "Ваша преобразованная матрица:" << endl;

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

{

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

{

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

}

fout << endl;

}

fout.close();

}

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

{

int doubleN = 2 \* n;

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

{

delete[] arr[i];

}

delete[] arr;

}

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

{

int\*\* arr = nullptr;

string way;

if (isInput)

{

cout << "Введите N: ";

n = parameterIn(MIN\_NUM);

arr = createMatrix(n);

arr = fillMatrix(arr, n);

}

else

{

way = wayToFile(1);

n = parameterInF(way);

arr = takeArrFromFile(way, arr, n);

}

return arr;

}

void finalMatrixOut(bool isOutput, int\*\* arr, int n)

{

string way;

if (isOutput)

{

cout << "Ваша матрица:" << endl;

writeFirstMatrix(arr, n);

cout << "Преобразованная матрица:" << endl;

writeNewMatrix(arr, n);

}

else

{

way = wayToFile(11);

writeMatrixF(way, arr, n);

writeNewMatrixF(way, arr, n);

}

}

int main()

{

setlocale(LC\_ALL, "Russian");

const int MIN\_NUM = 2;

bool isInput, isOutput;

int n;

int\*\* arr = nullptr;

string way;

cout << "Данная программа меняет подматрицы N вашей матрицы 2N местами" << endl;

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

isInput = consoleChoice();

arr = finalMatrixIn(MIN\_NUM, isInput, n);

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

isOutput = consoleChoice();

finalMatrixOut(isOutput, arr, n);

clearMemory(arr, n);

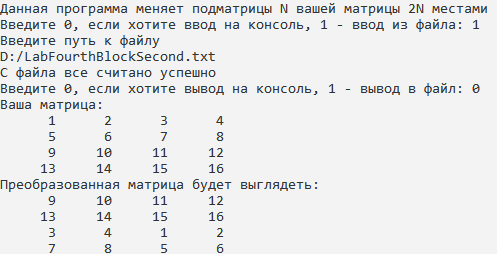
return 0;

}

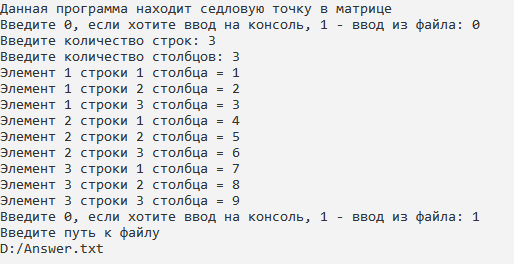
Код программы на **Java**:

import java.io.\*;  
import java.util.Scanner;  
public class LabFourthBlockSecond {  
 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 dimensionIn(final int MIN\_NUM){  
 boolean isIncorrect;  
 int n = 0;  
 do {  
 System.out.println("Введите размерность подматрицы n матрицы 2n: ");  
 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("Должно быть > 1");  
 isIncorrect = true;  
 }  
 } while (isIncorrect);  
 return n;  
 }  
  
 public static int[][] fillMatrix(int n) {  
 boolean isIncorrect;  
 int arr[][];  
 arr = new int[2\*n][2\*n];  
 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 String takeFileWay(int x) {  
 String way;  
 boolean isIncorrect = false;  
 do {  
 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, newN, elem;  
 boolean isIncorrect = false;  
 try (BufferedReader br = new BufferedReader(new FileReader(way))) {  
 Scanner scanFile = new Scanner(br);  
 n = Integer.parseInt(scanFile.nextLine()); // после этого finally  
 if (n < 2) throw new Exception ("Размерность подматрицы должна быть >   
 1");  
 newN = 4\*n\*n;  
 for (int i = 0; i < newN; i++) {  
 try {  
 elem = Integer.parseInt(scanFile.next());  
 } catch (Exception e) {  
 System.err.println("Проверьте правильность данных матрицы,   
 введенной в документе");  
 System.err.println(e.getMessage());  
 isIncorrect = true;  
 }  
 }  
 if (scanFile.hasNextLine()) {  
 isIncorrect = true;  
 System.err.println("Уберите лишние данные");  
 }  
 scanFile.close();  
 } catch (Exception e) {  
 System.err.println("Проверьте правильность данных, введенных в   
 документе");  
 System.err.println(e.getMessage());  
 isIncorrect = true;  
 }  
 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))) {  
 Scanner scanFile = new Scanner(br);  
 n = Integer.parseInt(scanFile.nextLine());  
 arr = new int[2\*n][2\*n];  
 for (int i = 0; i < arr.length; i++) {  
 for (int j = 0; j < arr[0].length; j++) {  
 arr[i][j] = Integer.parseInt(scanFile.next());  
 }  
 }  
 scanFile.close();  
 } catch (Exception e) {  
 System.err.println("Проверьте правильность данных, введенных в   
 документе");  
 }  
 return arr;  
 }  
  
 public static void writeFirstMatrix(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[][] convertMatrix(int[][] arr) {  
 int x;  
 for (int i = 0; i < (arr.length/2); i++) {  
 for (int j = 0; j < (arr[0].length / 2); j++) {  
 x = arr[i][j];  
 arr[i][j] = arr[i + arr.length / 2][j];  
 arr[i + arr.length / 2][j] = arr[i][j + arr.length / 2];  
 arr[i][j + arr.length / 2] = arr[i + arr.length / 2][j + arr.length /   
 2];  
 arr[i + arr.length / 2][j + arr.length / 2] = x;  
 }  
 }  
 return arr;  
 }  
  
 public static void writeNewMatrix(int[][] arr) {  
 arr = convertMatrix(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 void writeFirstMatrixF(String way, int[][] arr) {  
 try (PrintWriter pw = new PrintWriter(new FileWriter(way, false))) {  
 pw.write("Ваша первоначальная матрица:\n");  
 for (int i = 0; i < arr.length; i++) {  
 for (int j = 0; j < arr[0].length; j++) {  
 pw.printf("%5d", arr[i][j]); //используется printf PrintWriter  
 }  
 pw.write("\n"); //используется write FileWriter  
 }  
 } catch (Exception e) {  
 System.err.println("Ошибка файла");  
 System.out.println(e.getMessage());  
 }  
 }  
  
 public static void writeNewMatrixF(String way, int[][] arr) {  
 arr = convertMatrix(arr);  
 try (PrintWriter pw = new PrintWriter(new FileWriter(way, true))) {  
 pw.write("Ваша преобразованная матрица: \n");  
 for (int i = 0; i < arr.length; i++) {  
 for (int j = 0; j < arr[0].length; j++) {  
 pw.printf("%5d", arr[i][j]);  
 }  
 pw.write("\n");  
 }  
 } catch (Exception e) {  
 System.err.println("Ошибка файла");  
 System.out.println(e.getMessage());  
 }  
  
 }  
  
 public static int[][] finalMatrixIn(final int MIN\_NUM, boolean isInput) {  
 int [][] arr;  
 int n;  
 String way;  
 if (isInput) {  
 n = dimensionIn(MIN\_NUM);  
 arr = fillMatrix(n);  
 } else {  
 way = takeFileWay(1);  
 arr = takeArrFromFile(way);  
 }  
 return arr;  
 }  
  
 public static void finalMatrixOut(boolean isOutput, int[][] arr) {  
 String way;  
 if (isOutput) {  
 System.out.println("Ваша матрица: ");  
 writeFirstMatrix(arr);  
 System.out.println("Преобразованная матрица: ");  
 writeNewMatrix(arr);  
 } else {  
 way = takeFileWay(11);  
 writeFirstMatrixF(way, arr);  
 writeNewMatrixF(way, arr);  
 }  
 }  
  
 public static void main(String[] args) {  
 boolean isInput, isOutput;  
 int arr[][];  
 final int MIN\_NUM = 2;  
 System.out.println("Данная программа меняет подматрицы N вашей матрицы 2N   
 местами");  
 System.out.print("Введите 0, если хотите ввод на консоль, 1 — ввод из файла: ");  
 isInput = consoleChoice();  
 arr = finalMatrixIn(MIN\_NUM, isInput);  
 System.out.println("Введите 0, если хотите вывод на консоль, 1 – вывод в   
 файл: ");  
 isOutput = consoleChoice();  
 finalMatrixOut(isOutput, arr);  
 scan.close();  
 }  
}

Результат на **Delphi**:



Результат на **C++**:

****

Результат на **Java**:

