**УО «Белорусский государственный университет информатики и радиоэлектроники»**

**Кафедра ПОИТ**

**Отчёт по лабораторной работе №2.3**

**По предмету**

**Основы алгоритмизации и программирования**

**Вариант 6**

**Выполнил:**

**Гладкий М.Г.**

**Проверила:**

**Данилова Г.В.**

**Группа 851001**

**Минск 2018**

**Задание:**

Выполнить “прямой ход” в решении системы алгебраических уравнений методом Гаусса.

**Код Delphi 10:**

**program** Project7;

{$APPTYPE CONSOLE}

**Uses**

SysUtils;

**type**

TArray = **array**[0..255,0..255] **of** Real;

TArr = **array**[0..255] **of** Real;

**const**

MinSize = 2;

MaxSize = 255;

MinInt: Integer = -2147483646;

MaxInt: Integer = 2147483647;

**procedure** DirectCourse(Num, Unk: Byte; **var** Matrix: TArray; **var** Vector: TArr);

**var**

i, j, k, m: Byte;

Bufer: Real;

**begin**

**for** k := 0 **to** Unk - 1 **do**

**begin**

m := k;

**for** i := k + 1 **to** Num **do**

**if** abs(Matrix[m][k]) < abs(Matrix[i][k]) **then**

m := i;

**if** m <> k **then**

**begin**

**for** j := 0 **to** Unk **do**

**begin**

Bufer := Matrix[m][j];

Matrix[m][j] := Matrix[k][j];

Matrix[k][j] := Bufer;

**end**;

Bufer := Vector[m];

Vector[m] := Vector[k];

Vector[k] := Bufer;

**end**;

**for** i := k + 1 **to** Num **do**

**begin**

Bufer := Matrix[i][k] / Matrix[k][k];

Matrix[i][k] := 0;

Vector[i] := Vector[i] - Bufer \* Vector[k];

**if** Bufer <> 0 **then**

**for** j := k + 1 **to** Unk **do**

Matrix[i][j] := Matrix[i][j] - Bufer \* Matrix[k][j];

**end**;

**end**;

**end**;

**function** ChoiceInput(): Char;

**var**

Input: Char;

IsCorrect: Boolean;

**begin**

**repeat**

Readln(Input);

Input := UpCase(Input);

**if** (Input = 'Y')**or**(Input = 'N') **then**

IsCorrect := true

**else**

**begin**

IsCorrect := false;

Write('Incorrect input. Enter Y(Yes) or N(No): ');

**end**;

**until** IsCorrect;

ChoiceInput := Input;

**end**;

**procedure** ShowVectorFile(**var** NewFile: TextFile; Num: Byte; **var** Vector: TArr);

**var**

i: Byte;

**begin**

**for** i := 0 **to** Num **do**

Write(NewFile, Vector[i]:6:3,' ');

Writeln(NewFile);

**end**;

**procedure** ShowMatrixFile(**var** NewFile: TextFile; Num, Unk: Byte; **var** Matrix: TArray);

**var**

i, j: Byte;

**begin**

**for** i := 0 **to** Num **do**

**begin**

**for** j := 0 **to** Unk **do**

Write(NewFile, Matrix[i][j]:6:3,' ');

Writeln(NewFile);

**end**;

**end**;

**procedure** ShowAnswerFile(**var** NewFile: TextFile; Num, Unk: Byte; **var** Matrix: TArray;

**var** Vector: TArr);

**begin**

Writeln(NewFile, 'Matrix:');

ShowMatrixFile(NewFile, Num, Unk, Matrix);

Writeln(NewFile, 'Vector:');

ShowVectorFile(NewFile, Num, Vector);

**end**;

**procedure** ShowVector(Num: Byte; **var** Vector: TArr);

**var**

i: Byte;

**begin**

**for** i := 0 **to** Num **do**

Write(Vector[i]:6:3,' ');

Writeln;

**end**;

**procedure** ShowMatrix(Num, Unk: Byte; **var** Matrix: TArray);

**var**

i, j: Byte;

**begin**

**for** i := 0 **to** Num **do**

**begin**

**for** j := 0 **to** Unk **do**

Write(Matrix[i][j]:6:3,' ');

Writeln;

**end**;

**end**;

**procedure** OutputConsole(Num, Unk: Byte; **var** Matrix: TArray; **var** Vector: TArr);

**begin**

Writeln('Matrix:');

ShowMatrix(Num, Unk, Matrix);

Writeln('Vector:');

ShowVector(Num, Vector);

**end**;

**procedure** OutputFile(Num, Unk: Byte; **var** Matrix: TArray; **var** Vector: TArr);

**var**

IsCorrect: Boolean;

NewFile: TextFile;

NameOfFile: String;

**begin**

Write('Enter the name of file (.txt): ');

IsCorrect := false;

**repeat**

Readln(NameOfFile);

**if** (**not** FileExists(NameOfFile)) **then**

Write('File does not exist. Try again: ')

**else**

**begin**

IsCorrect := true;

Assign(NewFile, NameOfFile);

Write('Would you like to rewrite the file? Enter Y(Yes) or N(No): ');

**if** ChoiceInput = 'Y' **then**

**try**

Rewrite(NewFile);

ShowAnswerFile(NewFile, Num, Unk, Matrix, Vector);

**except**

Write('Access is not allowed. Try again: ');

IsCorrect:= false;

**end**

**else**

**try**

Append(NewFile);

ShowAnswerFile(NewFile, Num, Unk, Matrix, Vector);

**except**

Write('Access is not allowed. Try again: ');

IsCorrect:= false;

**end**;

**end**;

**until** IsCorrect;

Close(NewFile);

**end**;

**function** CheckInput(Min, Max: Integer): Integer;

**var**

Number: Integer;

IsCorrect: Boolean;

**begin**

IsCorrect := false;

**repeat**

**try**

Readln(Number);

**if** (Number >= Min) **and** (Number <= Max) **then**

IsCorrect := true

**else**

Writeln('Enter number from interval [', Min, '..', Max, ']: ');

**except**

Writeln('Check entered data. Enter number from interval [', Min, '..',

Max, ']: ');

**end**;

**until** IsCorrect;

CheckInput := Number;

**end**;

**procedure** GetDataConsole(**var** Num, Unk: Byte; **var** Matrix: TArray; **var** Vector: TArr);

**var**

i, j: Byte;

**begin**

Writeln('Enter number of equations[', MinSize,'..', MaxSize,']: ');

Num := CheckInput(MinSize, MaxSize);

Writeln('Enter number of unknowns[', MinSize,'..', MaxSize,']: ');

Unk := CheckInput(MinSize, MaxSize);

Dec(Num);

Dec(Unk);

Writeln('Enter your matrix:');

**for** i := 0 **to** Num **do**

**for** j := 0 **to** Unk **do**

**begin**

Write('Element[',i + 1,'][',j + 1,'] = ');

Matrix[i][j] := CheckInput(MinInt, MaxInt);

**end**;

Writeln('Enter values of equations:');

**for** i := 0 **to** Num **do**

**begin**

Write('Element[',i + 1,'] = ');

Vector[i] := CheckInput(MinInt, MaxInt);

**end**;

Writeln('Your input is:');

OutputConsole(Num, Unk, Matrix, Vector);

**end**;

**function** CheckInputFile(Min, Max: Integer; **var** MyFile: TextFile): Boolean;

**var**

IsCorrect: Boolean;

Number: Integer;

**begin**

IsCorrect := true;

**while** (**not** SeekEof(MyFile)) **and** (IsCorrect) **do**

**try**

Readln(MyFile, Number);

**except**

IsCorrect := false;

**end**;

CheckInputFile := IsCorrect;

**end**;

**function** ReadFromFile(**var** MyFile: TextFile; **var** Num, Unk: Byte; **var** Matrix: TArray;

**var** Vector: TArr): Boolean;

**var**

i, j: Byte;

**begin**

**if** CheckInputFile(MinInt, MaxInt, MyFile) **then**

**begin**

reset(MyFile);

i := 0;

**while not** Eof(MyFile) **do**

**begin**

j := 0;

**while not** Eoln(MyFile) **do**

**begin**

Read(MyFile, Matrix[i][j]);

inc(j);

**end**;

Readln(MyFile);

inc(i);

**end**;

Num := i - 1;

Unk := j - 2;

Writeln('Matrix:');

**for** i := 0 **to** Num **do**

**begin**

Vector[i] := Matrix[i][Unk + 1];

**for** j := 0 **to** Unk **do**

Writeln('Element[', i + 1, '][', j + 1, '] = ', Matrix[i,j]:6:3);

**end**;

Writeln('Vector:');

**for** i := 0 **to** Num **do**

Writeln('Element[', i + 1, '] = ', Vector[i]:6:3);

**end**

**else**

**begin**

CloseFile(MyFile);

ReadFromFile := false;

Write('Check entered data. Enter number from interval ', MinInt, '..', MaxInt,

'. Try Again: ');

**end**;

**end**;

**procedure** GetDataFile(**var** Num, Unk: Byte; **var** Matrix: TArray; **var** Vector: TArr);

**var**

IsCorrect: Boolean;

NameOfFile: String;

MyFile: TextFile;

**begin**

Write('Enter file name(.txt): ');

**repeat**

Readln(NameOfFile);

**if** (**not** FileExists(NameOfFile)) **then**

**begin**

WriteLn('File does not exist. Try again: ');

IsCorrect := false;

**end**

**else**

**begin**

AssignFile(MyFile, NameOfFile);

reset(MyFile);

**if** SeekEof(MyFile) **then**

**begin**

Writeln('File is empty. Try again: ');

IsCorrect := false;

**end**

**else**

IsCorrect := ReadFromFile(MyFile, Num, Unk, Matrix, Vector);

**end**;

**until** IsCorrect;

CloseFile(MyFile);

**end**;

**procedure** Main();

**var**

Matrix: TArray;

Vector: TArr;

Num, Unk: Byte;

**begin**

Writeln('This program performs the Direct course of the Gauss method for the

matrix');

Write('Would you like to use File input instead of Console input? Enter Y(Yes) or

N(No): ');

**if** ChoiceInput = 'Y' **then**

GetDataFile(Num, Unk, Matrix, Vector)

**else**

GetDataConsole(Num, Unk, Matrix, Vector);

DirectCourse(Num, Unk, Matrix, Vector);

Write('Would you like to write down the answer to File? Enter Y(Yes) or N(No): ');

**if** ChoiceInput = 'Y' **then**

OutputFile(Num, Unk, Matrix, Vector)

**else**

OutputConsole(Num, Unk, Matrix, Vector);

Writeln('Press "Enter" to exit.');

Readln;

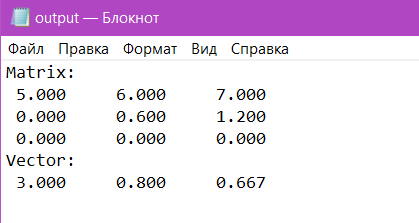
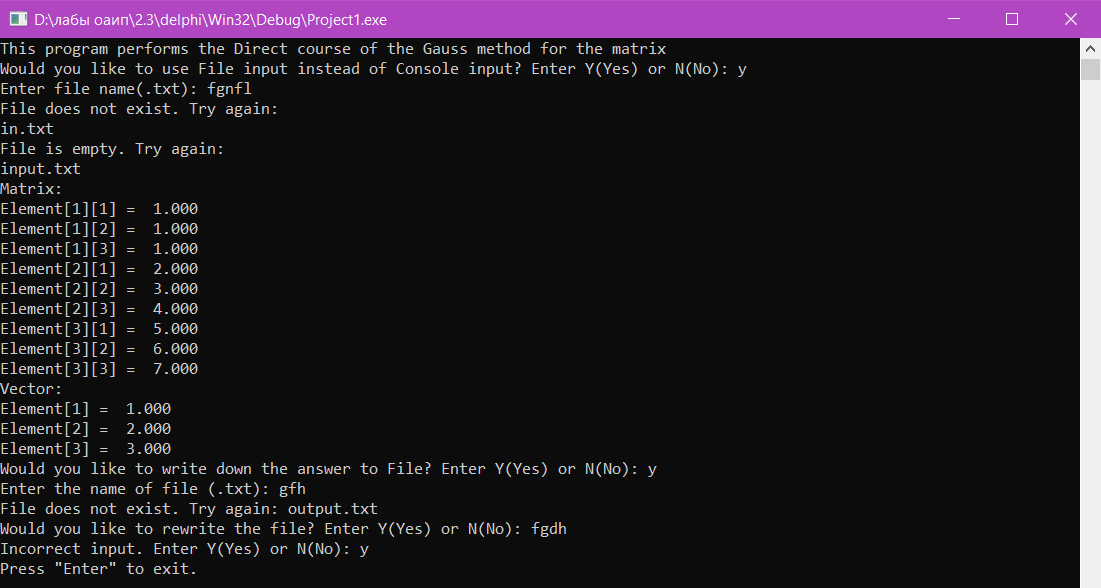
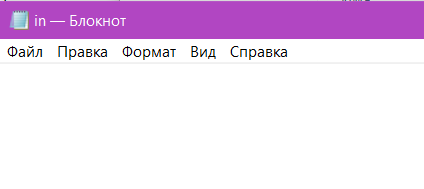
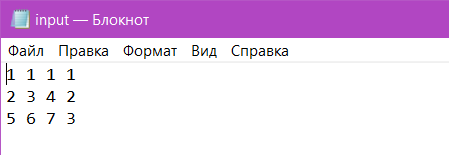
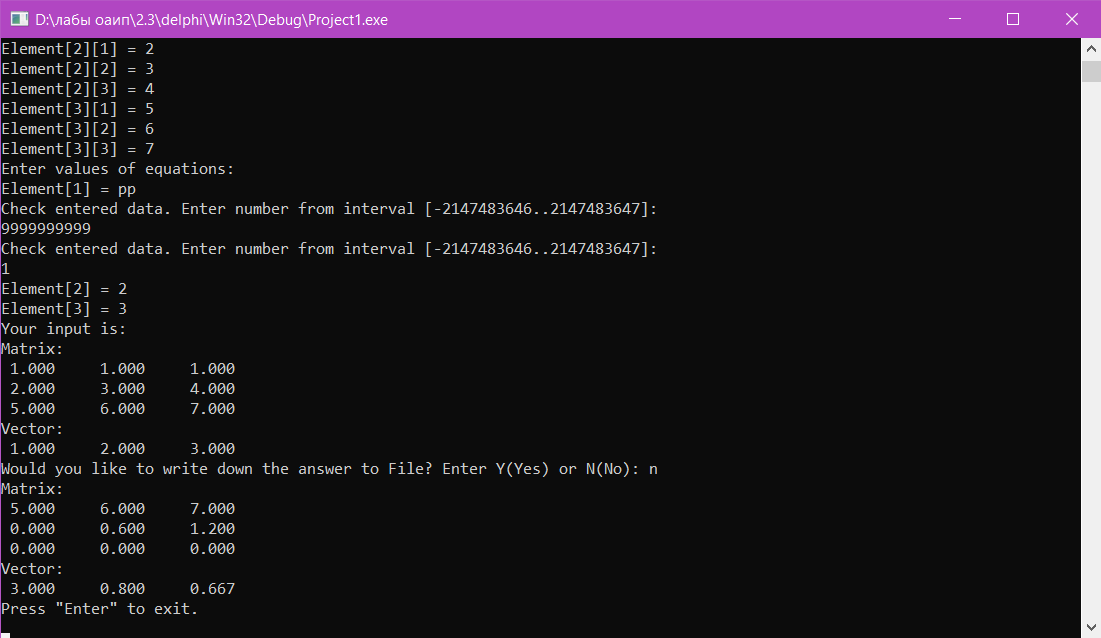
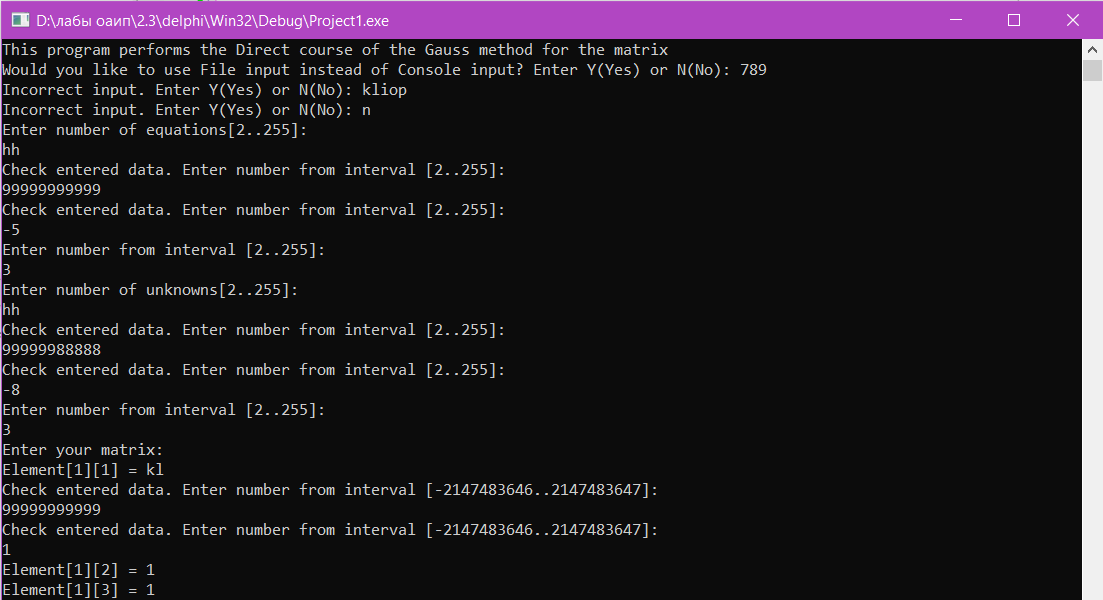
**end**;

**begin**

Main;

**end**.

**Скриншоты:**



**Блок-схема:**