УО «Белорусский государственный университет информатики и

радиоэлектроники»

Кафедра ПОИТ

Отчет по лабораторной работе № 2.3

по предмету

Основы Алгоритмизации и Программирования

Вариант 4

Выполнил

Воривода М.А.

Проверила

Данилова Г.В.

Группа:

951007

Минск 2019

**Задание**

Дана квадратная матрица A порядка n. Проверить, является ли матрица единичной.

**Код программы**

**(Delphi)**

program Goal3;

uses

SysUtils;

type

ArrayOfString = array of String;

function IsMatrixIdentity(A: array of ArrayOfString): Boolean;

var

Row, Col: Integer;

Res: Boolean;

begin

Res := true;

for Row := 0 to length(A) - 1 do

begin

col := 0;

while (Col < length(A[Row])) and (Res) do

begin

if Col = Row then

if A[Row][Col] = '1' then

inc(Col)

else

Res := false

else if A[Row][Col] = '0' then

inc(Col)

else

Res := false;

end;

end;

Result := Res;

end;

function Split(S: String; N: Integer): ArrayOfString;

var

A: ArrayOfString;

I: Integer;

begin

SetLength(A, N);

I := 0;

while (pos(' ', S) > 0) and (I < N + 1) do

begin

A[I] := Copy(S, 1, (pos(' ', S) - 1));

delete(S, 1, pos(' ', S));

inc(I);

end;

A[I] := S;

Result := A;

end;

procedure Main();

var

N, I, J: Integer;

Reader: String;

A: array of ArrayOfString;

InF, OutF: TextFile;

Valid: Boolean;

begin

Valid := false;

repeat

try

WriteLn('Enter input file directory: ');

ReadLn(Reader);

AssignFile(InF, Reader);

Reset(InF);

Valid := true;

except

WriteLn('File not found');

end;

until Valid;

WriteLn('Enter output file directory: ');

ReadLn(Reader);

AssignFile(OutF, Reader);

Rewrite(OutF);

Valid := false;

try

begin

ReadLn(InF, Reader);

N := StrToInt(Reader);

SetLength(A, N, N);

for I := 0 to N - 1 do

begin

ReadLn(InF, Reader);

A[I] := Split(Reader, N);

end;

Valid := true;

end;

except

WriteLn(OutF, 'INPUT ERROR');

WriteLn('INPUT ERROR');

end;

if Valid then

begin

WriteLn;

WriteLn(OutF, 'Is entered matrix identity?');

WriteLn('Is entered matrix identity?');

WriteLn;

for I := 0 to length(A) - 1 do

begin

for J := 0 to length(A) - 2 do

begin

Write(OutF, A[I][J], ', ');

Write(A[I][J], ', ');

end;

WriteLn(OutF, A[I][J]);

WriteLn(A[I][J]);

end;

WriteLn(outF);

WriteLn;

if IsMatrixIdentity(A) then

begin

WriteLn(OutF, 'YES');

WriteLn('YES')

end

else

begin

WriteLn(OutF, 'NO');

WriteLn('NO')

end;

end;

CloseFile(InF);

CloseFile(OutF);

end;

begin

Main();

ReadLn;

end.

**Код программы**

**(C)**

#include <stdio.h>

#include <stdbool.h>

bool isMatrixIdentity (int N, int (\*A)[N]) {

int col;

for ( int row = 0; row < N; row++ ) {

col = 0;

while ( col < N) {

if ( col == row ) {

if (A[row][col] == 1) {

col++;

} else {

return false;

}

} else {

if (A[row][col] == 0) {

col++;

} else {

return false;

}

}

}

}

return true;

}

void main() {

FILE \*inF = fopen("D:\\University\\OAiP\\LAB2\\C\\Goal

3\\input.txt", "r");

FILE \*outF = fopen("D:\\University\\OAiP\\LAB2\\C\\Goal 3

\\output.txt", "w");

int N = 0;

fscanf(inF, "%d", &N);

int A[N][N];

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

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

fscanf(inF, "%d ", &A[i][j]);

}

}

fprintf(outF,"Is entered matrix identity?\n");

printf("Is entered matrix identity?\n");

int j;

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

for (j = 0; j < N - 1; j++) {

fprintf(outF, "%d, ", A[i][j]);

printf("%d, ", A[i][j]);

}

fprintf(outF, "%d\n", A[i][j]);

printf("%d\n", A[i][j]);

}

if (isMatrixIdentity(N, A)) {

fprintf(outF,"YES");

printf("YES");

} else {

fprintf(outF, "NO");

printf("NO");

}

fclose(inF);

fclose(outF);

}

**Код программы**

**(Java)**

import java.io.\*;

import java.util.Scanner;

public class Main {

public static void main(String[] args) throws Exception {

Scanner in = new Scanner(System.in);

FileReader inF = new FileReader("input.txt");

FileWriter outF = new FileWriter("output.txt");

int N = 0;

boolean inValid = true;

do {

try {

System.out.println("Enter input file directory:

");

inF = new FileReader(in.nextLine());

inValid = false;

} catch (Exception e) {

System.out.println("File not found");

}

} while (inValid);

Scanner fileScanner = new Scanner(inF);

try {

N = Integer.parseInt(fileScanner.nextLine());

inValid = true;

} catch (Exception e) {

outF.write("INPUT ERROR");

System.out.println("INPUT ERROR");

}

String[][] A = new String[N][N];

try {

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

A[i] = split(fileScanner.nextLine(), N);

}

inValid = true;

} catch(Exception e) {

outF.write("INPUT ERROR");

System.out.println("INPUT ERROR");

}

int j;

if (inValid) {

outF.write("Is entered matrix identity?\n");

System.out.println("Is entered matrix identity?");

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

for (j = 0; j < N - 1; j++) {

outF.write(A[i][j] + ", ");

System.out.print(A[i][j] + ", ");

}

outF.write(A[i][j] + "\n");

System.out.print(A[i][j] + "\n");

}

if (isMatrixIdentity(A)) {

outF.write("YES");

System.out.print("YES");

} else {

outF.write("NO");

System.out.print("NO");

}

}

inF.close();

outF.close();

}

public static String[] split(String str, int Size) {

String[] A = new String[Size];

int i = 0;

while (str.indexOf(" ") > 0 && i < Size) {

A[i] = str.substring(0, str.indexOf(" "));

str = str.replaceFirst(A[i] + " ", "");

i++;

}

A[i] = str;

return A;

}

public static boolean isMatrixIdentity (String[][] A) {

int col;

for ( int row = 0; row < A.length; row++ ) {

col = 0;

while ( col < A[row].length) {

if ( col == row ) {

if (A[row][col].equals("1")) {

col++;

} else {

return false;

}

} else {

if (A[row][col].equals("0")) {

col++;

} else {

return false;

}

}

}

}

return true;

}

}

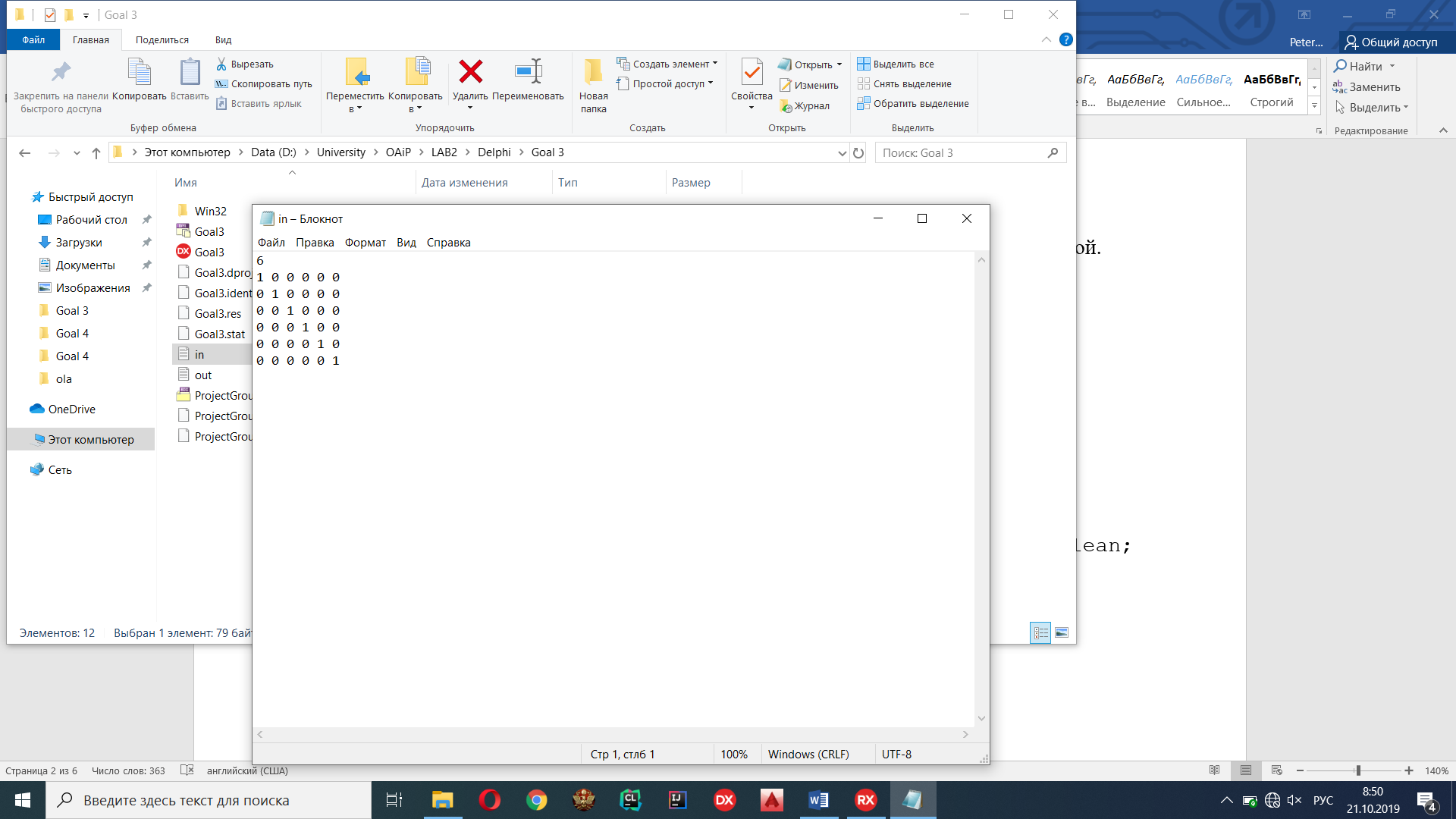
**Блок-схема**



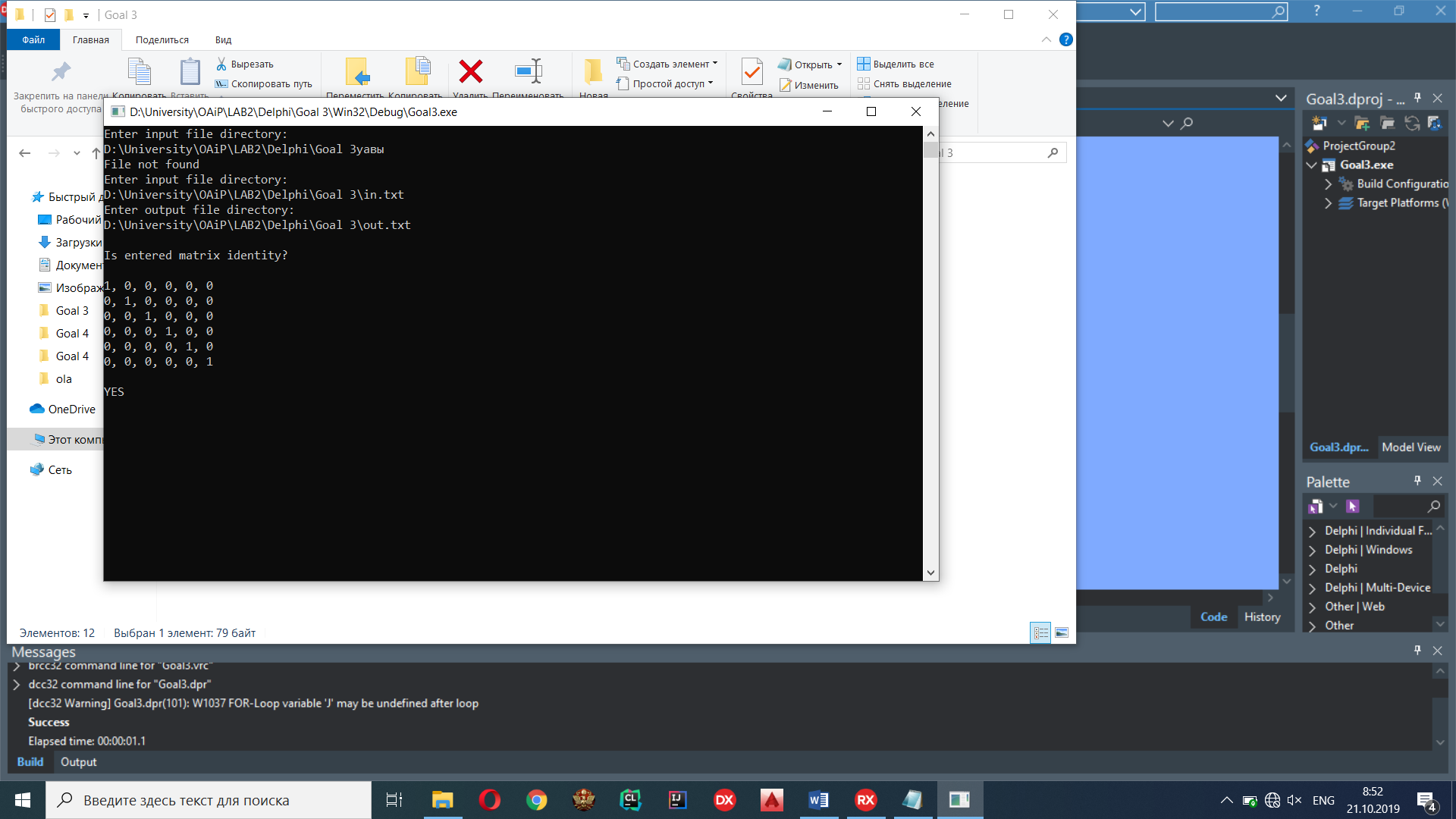
**Работа программы**

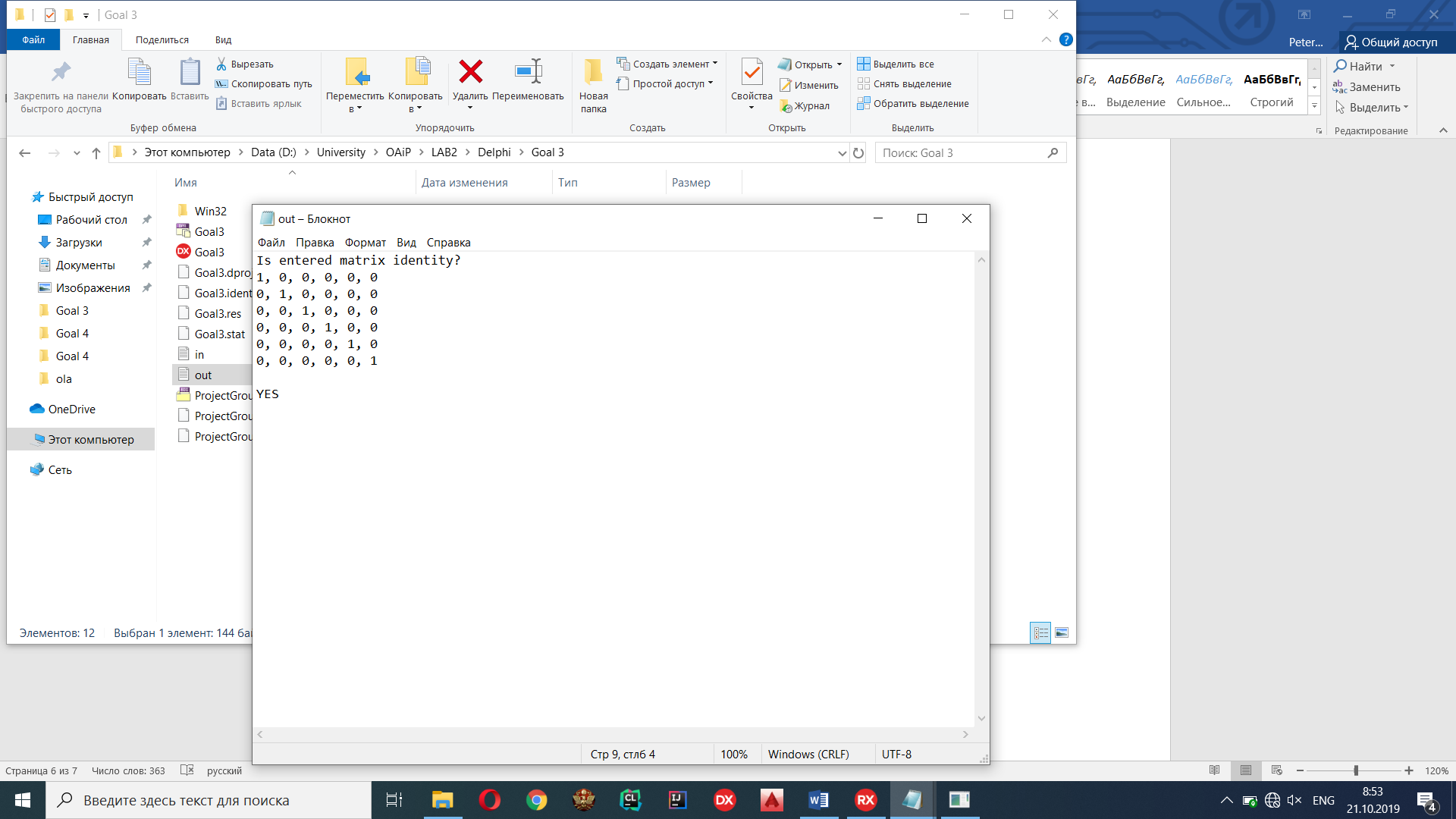
**Delphi**

Input:



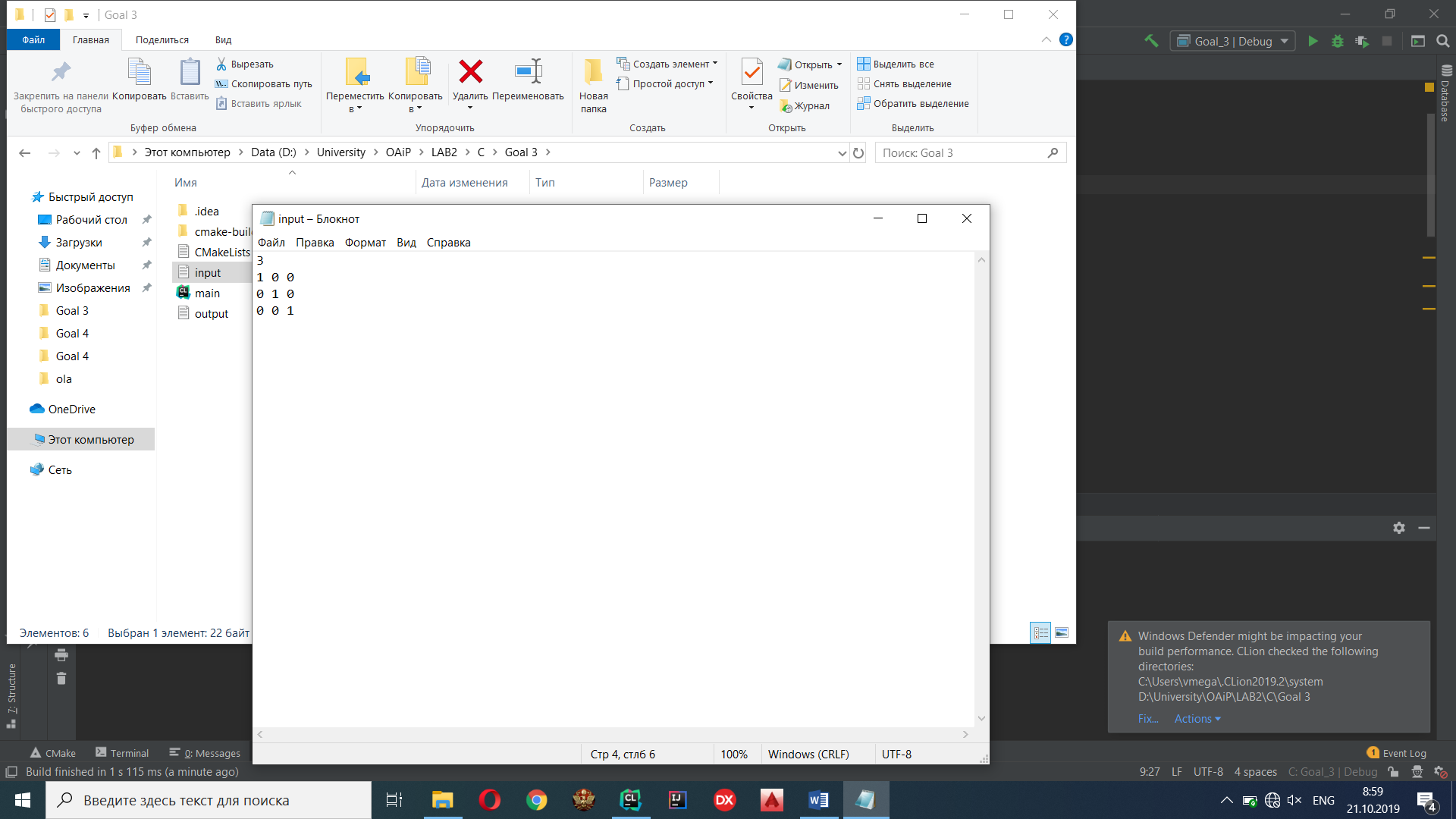
Output:



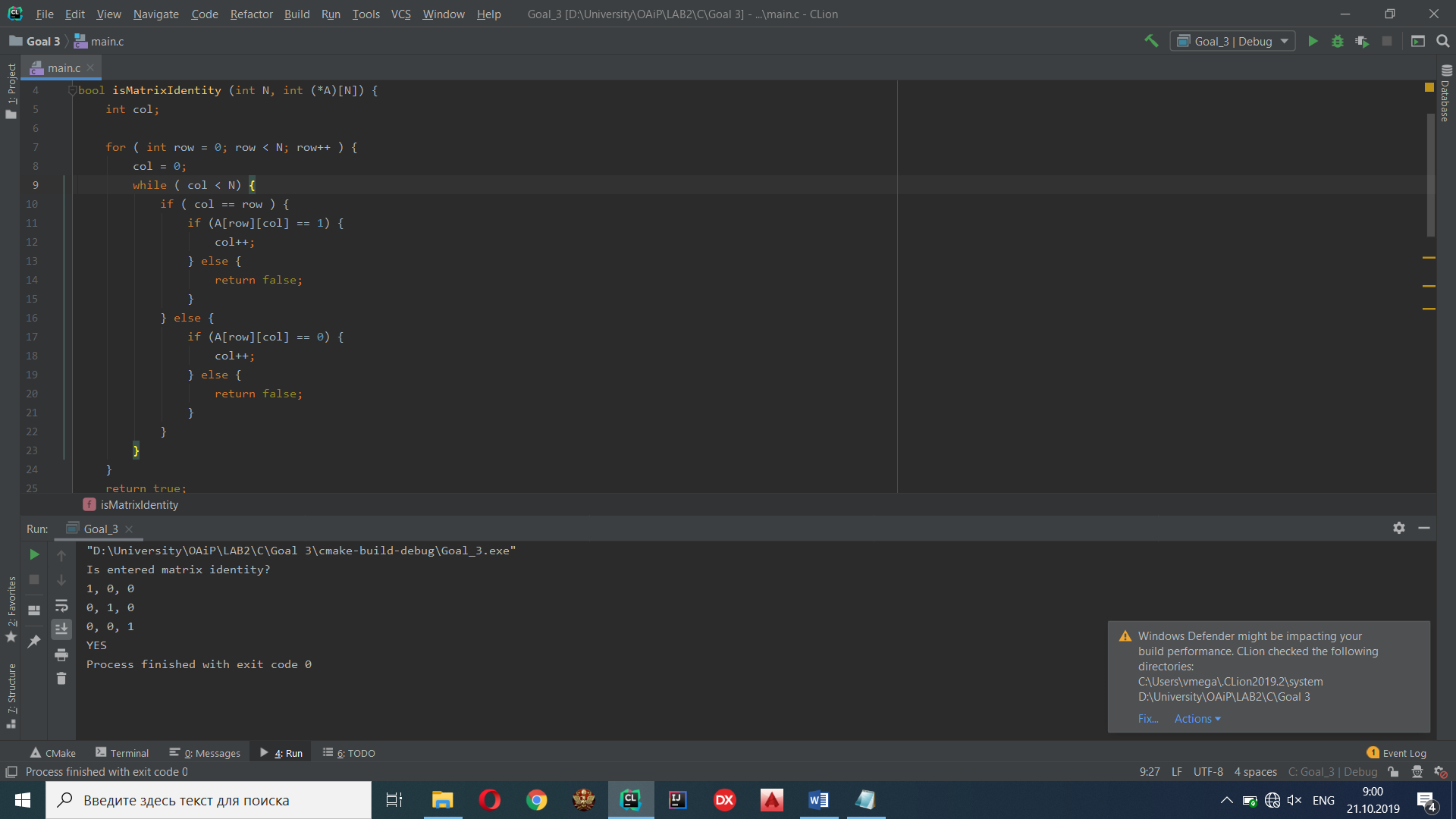


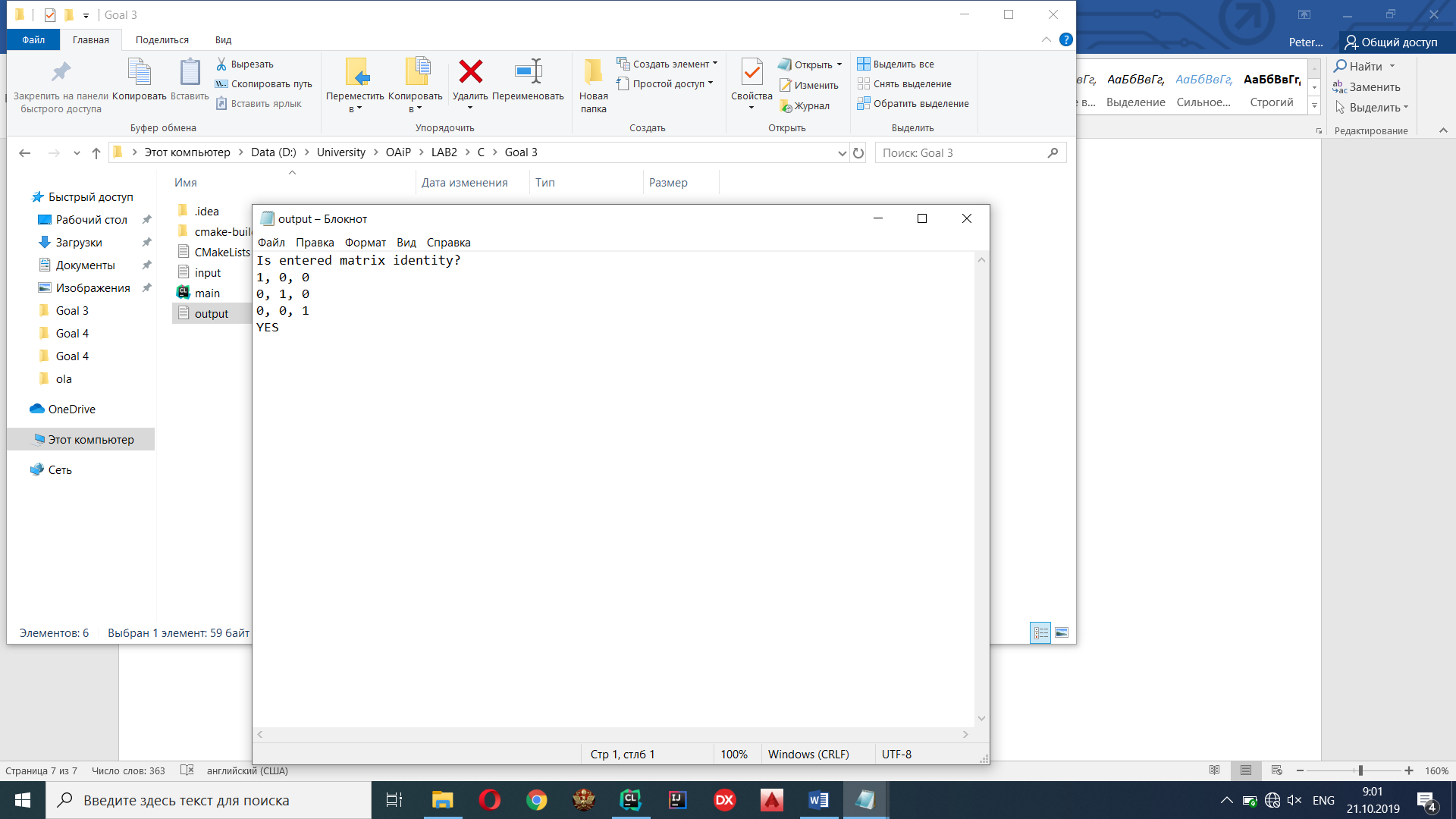
**C**

Input:



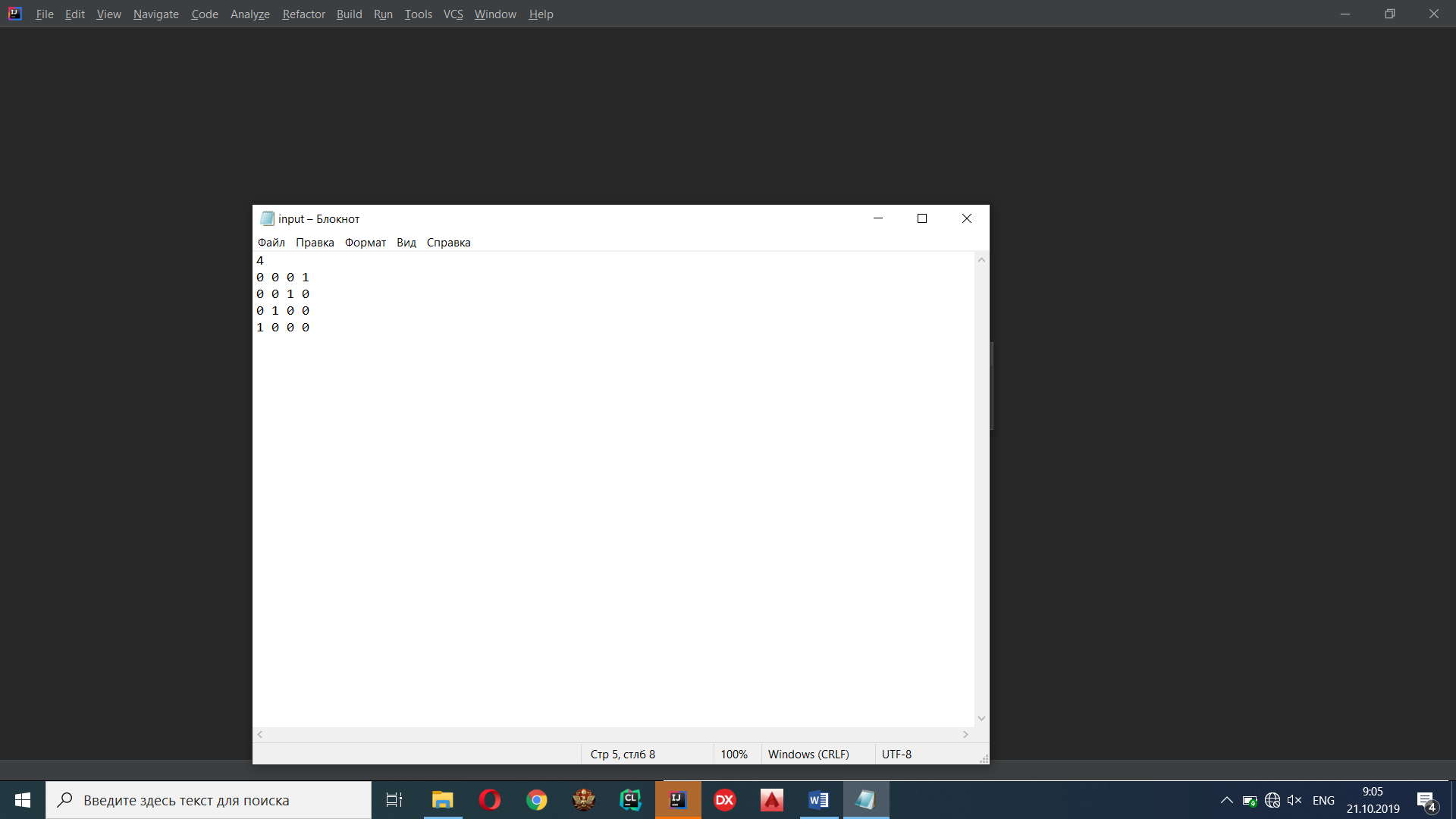
Output:





**Java**

Input:



Output:

