Министерство науки и высшего образования Российской Федерации

Пензенский государственный университет

Кафедра «Вычислительная техника»

**ОТЧЕТ**

по лабораторной работе №5

по курсу «Логика и основы алгоритмизации в инженерных задачах»

### на тему «Определение характеристик графов»

**Выполнили студенты**

**группы 21вв1.1:**

Федулов Е.Д.

Жбанников Д.Н.

**Приняли:**

Юрова О.В.

Акифьев И.В.

Пенза, 2022

**Цель работы:** научиться определять характеристики графов, строить матрицы смежности и инцидентности для графов.

**Задания:**

**Задание 1**

1. Сгенерируйте (используя генератор случайных чисел) матрицу смежности

для неориентированного графа G. Выведите матрицу на экран.

2. Определите размер графа G, используя матрицу смежности графа.

3. Найдите изолированные, концевые и доминирующие вершины.

**Задание 2\***

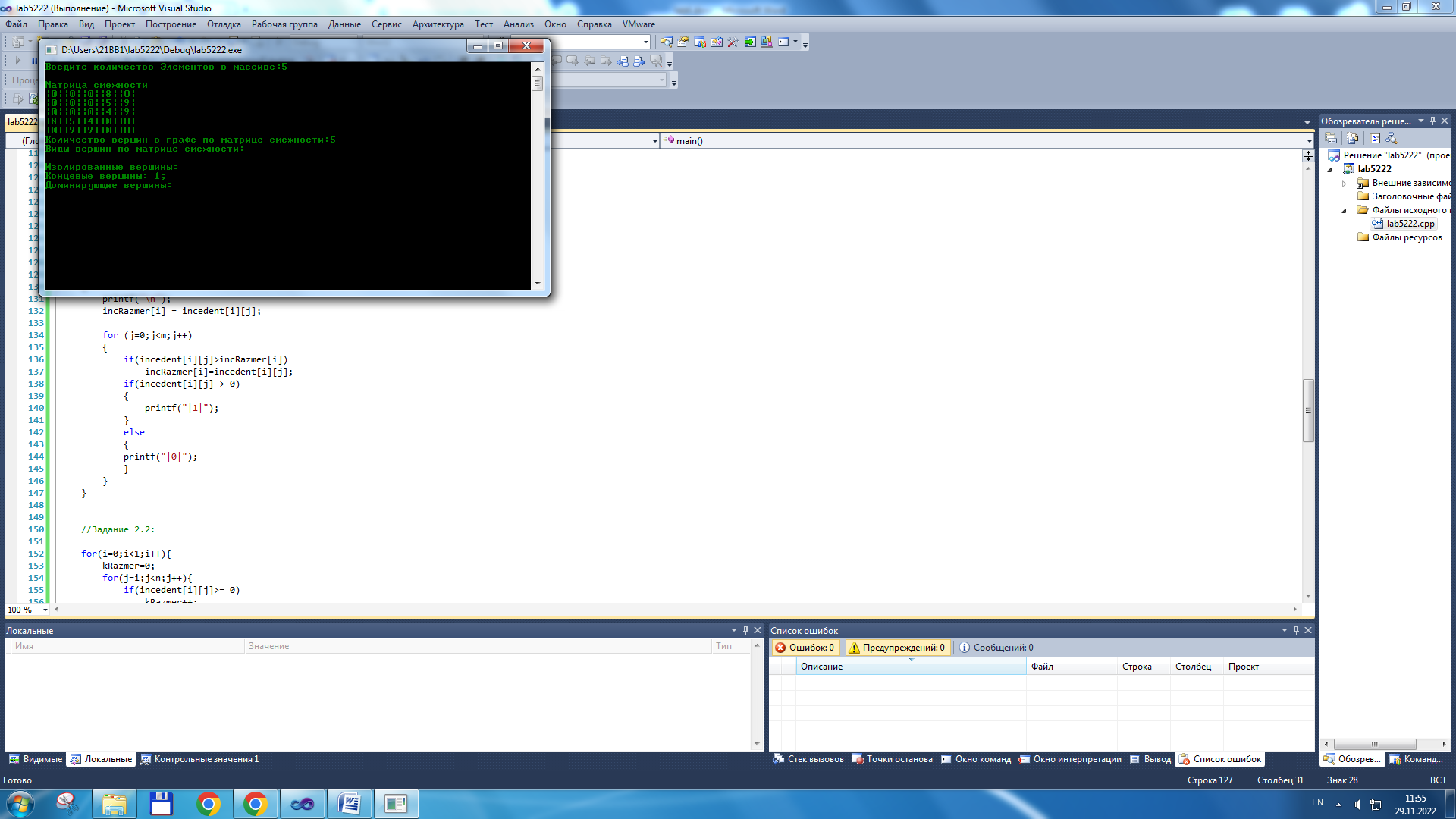
1. Постройте для графа G матрицу инцидентности.

2. Определите размер графа G, используя матрицу инцидентности графа.

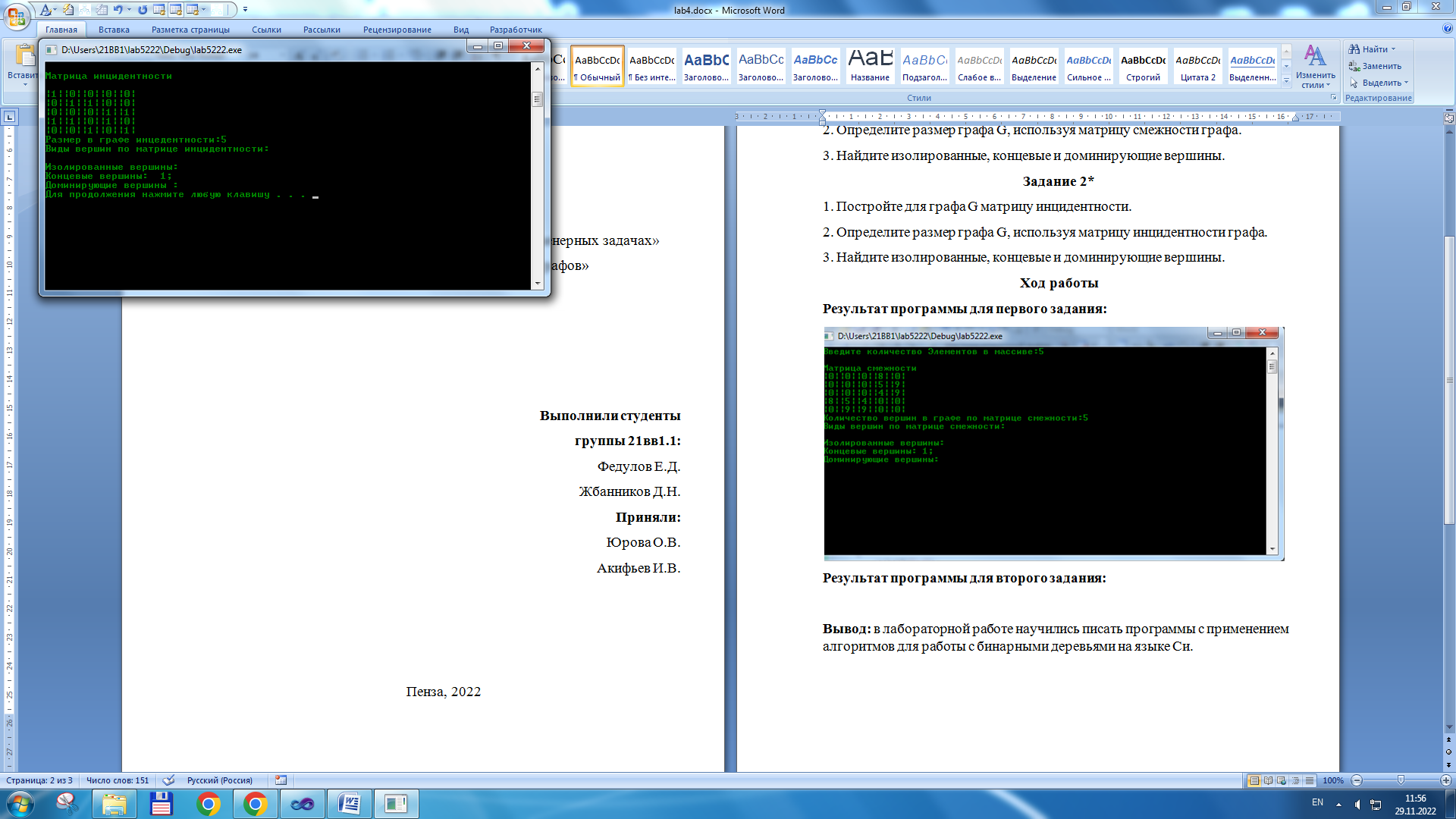
3. Найдите изолированные, концевые и доминирующие вершины.

**Ход работы**

**Результат программы для первого задания:**

****

**Результат программы для второго задания:**

****

**Вывод:** научились определять характеристики графов, строить матрицы смежности и инцидентности для графов.

**Общий листинг:**

#include<stdio.h>

#include<stdlib.h>

#include<conio.h>

#include<locale.h>

#include<Windows.h>

#include<malloc.h>

#include<time.h>

intmain()

{

setlocale(LC\_ALL,"rus");

int\*\*M;

constintrazmer=100;

intm,q,n,i,j,kRazmer=0,k=0,incRazmer[razmer];

//

char\*incedent[razmer];

printf("Введите количество Элементов в массиве:");

scanf\_s("%d",&n);

//random graf

srand((unsigned)time(NULL));

M=(int\*\*)malloc(n\*sizeof(int\*));

printf\_s("\nМатрица смежности");

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

{

M[i]=(int\*)malloc(n\*sizeof(int));

}

//Задание 1.1:

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

{

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

if(rand()%2){

M[i][j]=rand()%10;

}

else

M[i][j]=0;

if(i==j){

M[i][j]=0;

}

M[j][i]=M[i][j];

}

}

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

printf\_s("\n");

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

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

}

}

//Задание 1.2:

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

for(j=i;j<n;j++){

if(M[i][j]>0){

kRazmer+=1;

}

}

}

printf("\nКоличество вершин в графе по матрице смежности:%d",kRazmer);

//Задание 1.3:

printf("\nВиды вершин по матрице смежности: ");

printf("\n");

printf("\nИзолированные вершины: ");

for(inti=0;i<n;i++)

{

intform=0;

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

{

if(M[i][j]!=0)form++;

}

if(form==0)

printf("%2d;",i+1);

}

printf("\nКонцевые вершины:");

for(inti=0;i<n;i++)

{

intform=0;

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

{

if(M[i][j]!=0)form++;

}

if(form==1)

printf("%2d;",i+1);

}

printf("\nДоминирующие вершины: ");

for(inti=0;i<n;i++)

{

intform=0;

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

{

if(M[i][j]!=0)form++;

}

if(form==n-1)

printf("%2d;",i+1);

}

//Задание 2.1:

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

for(j=i+1;j<n;j++){

if(M[i][j])

m++;// Нашли число вершин

}

}

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

incedent[i]=(char\*)malloc(m);

memset(incedent[i],0,m);

}

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

for(j=i+1;j<n;j++){

if(M[i][j]){

incedent[i][q]=incedent[j][q]=M[i][j];

q++;}

}

}

printf("\n\n Матрица инцидентности ");

printf("\n");

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

{

printf("\n");

incRazmer[i]=incedent[i][j];

for(j=0;j<m;j++)

{

if(incedent[i][j]>incRazmer[i])

incRazmer[i]=incedent[i][j];

if(incedent[i][j]>0)

{

printf("|1|");

}

else

{

printf("|0|");

}

}

}

//Задание 2.2:

for(i=0;i<1;i++){

kRazmer=0;

for(j=i;j<n;j++){

if(incedent[i][j]>=0)

kRazmer++;

}

}

printf("\nРазмер в графе инцедентности:%d",kRazmer);

//Задание 2.3:

printf("\nВиды вершин по матрице инцидентности: ");

printf("\n");

printf("\nИзолированные вершины: ");

for(inti=0;i<n;i++)

{

intform=0;

for(j=0;j<m;j++)

{

if(incedent[i][j]!=0)form++;

}

if(form==0)

printf(" %2d;",i+1);

}

printf("\nКонцевые вершины:");

for(inti=0;i<n;i++)

{

intform=0;

for(j=0;j<m;j++)

{

if(incedent[i][j]!=0)form++;

}

if(form==1)

printf(" %2d;",i+1);

}

printf("\nДоминирующие вершины :");

for(inti=0;i<n;i++)

{

intform=0;

for(j=0;j<m;j++)

{

if(incedent[i][j]!=0)form++;

}

if(form==n-1)

printf("%2d,",i+1);

}

printf("\n");

system("Pause");

return0;

}