Министерство образования Российской Федерации

Пензенский государственный университет

Кафедра «Вычислительная техника»

**ОТЧЕТ**

по лабораторной работе №4

по курсу «Логика и основы алгоритмизации в инженерных задачах»

на тему «Обход графа в глубину»

Выполнили:

студенты группы 19ВВ2

Аниськин Н.В.

Приняли:

старший преподаватель

### Название

Программа, содержащая в себе функцию обхода графа в глубину (DFS).

### Цель работы

Научиться работать с рекурсивной функцией обхода графа в глубину.

**Лабораторное задание**

### **Задание 1**

1. Сгенерируйте (используя генератор случайных чисел)матрицу смежности для неориентированного графа *G*. Выведите сгенерированную матрицу на экран.

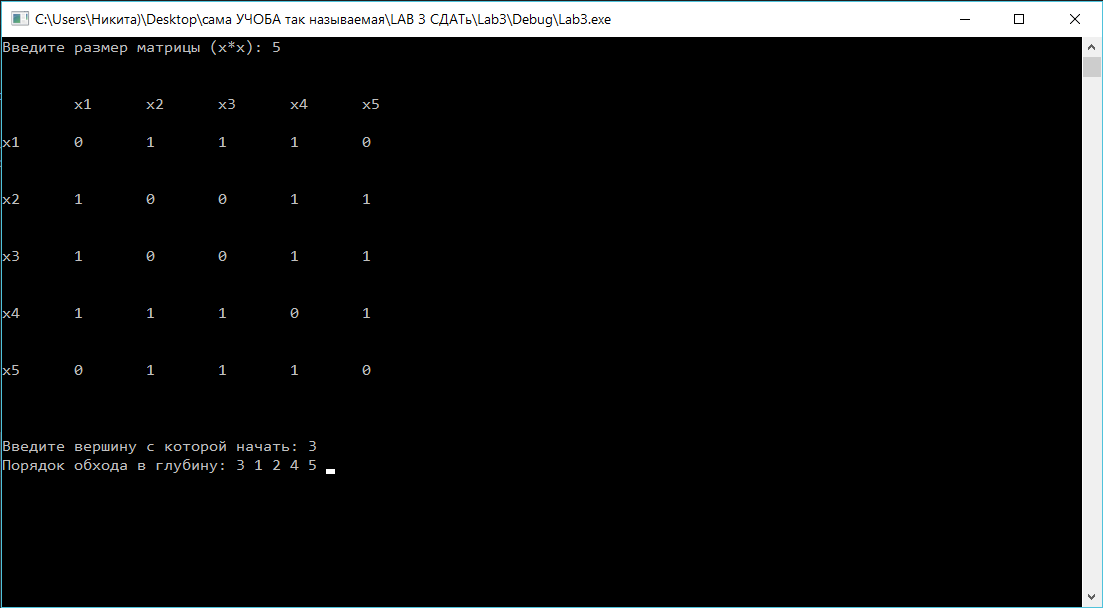
2.Для сгенерированного графа осуществите процедуру обхода в глубину, реализованную в соответствии с приведённым описанием.

### **Пояснительный текст к программе**

Программа представляет из код, состоящий из основной функции и функции обхода графа в глубину.

1. Генерируется матрица смежности (с псевдослучайными значениями), выводится в виде графа.
2. С клавиатуры вводится вершина, с которой будет выполнен обход графа.
3. Все вершины помечаются как непосещенные.
4. Выполняется функция обхода графа в глубину, её шаги выводятся на экран.

### **Результаты работы программы**

****

**Рисунок 1 — Результаты работы программы**

### **Выводы**

В ходе выполнения лабораторной работы была разработана программа, выполняющая унарные и бинарные операции над графами в матричном представлении

Получили опыт в создании проектов в среде Microsoft Visual Studio, научились писать и отлаживать линейные программы на языке Си.

### **Листинг программы**

### Файл «Source.cpp»

#include <stdio.h>

#include <conio.h>

#include <time.h>

#include <locale.h>

#include <stdlib.h>

using namespace std;

int i, j, m;

bool\* visited = new bool[m];

int\*\* graph;

void DFS(int st)

{

int k;

printf("%d ", st + 1);

visited[st] = true;

for (k = 0; k <= m; k++){

if ((graph[st][k] != 0) && (!visited[k]))

DFS(k);

}

}

void main()

{

setlocale(LC\_ALL, "Rus");

printf("Введите размер матрицы (x\*x): ");

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

printf("\n\n");

////создание и генерация графа

graph = new int\*[m];

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

graph[i] = new int[m];

}

srand(time(NULL));

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

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

graph[i][j] = graph[j][i] = rand() % 2;

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

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

if (j == i)

{

graph[j][i] = 0;

}

}

}

////////////////////////// ВЫВОД графа ////////

printf(" \t");

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

{

printf("x%d\t", j + 1);

}

printf("\n\n");

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

{

printf("x%d\t", i + 1);

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

printf("%d\t", graph[i][j]);

printf("\n\n\n");

}

/////////////////// DFS

int vershina;

printf("\nВведите вершину с которой начать: ");

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

while (vershina > m) {

printf("\nТакой вершины не существует\n");

printf("\nВведите вершину с которой начать: ");

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

}

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

visited[i] = false; // помечаем как !visited все вершины

}

printf("Порядок обхода в глубину: ");

DFS(vershina - 1);

\_getch();

}