Министерство науки и высшего образования РоссийскойФедерации

Пензенский государственный университет

Кафедра «Вычислительная техника»

**ОТЧЕТ**

по лабораторной работе №5

по теме «Унарные и бинарные операции над графами»

Выполнили:

Студенты группы 21ВВ2

Изосин А.А.

Волдырев В.С.

Принял:

Митрохин М.А.

Юрова О.В.

Пенза, 2022

**Цель работы:**

Исследование унарных, бинарных операций над графами и приобретение практических навыков решения задач с использованием основ теории графов.

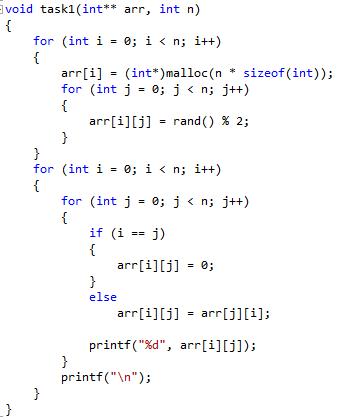
**Лабораторные работы:**

**Задание:**

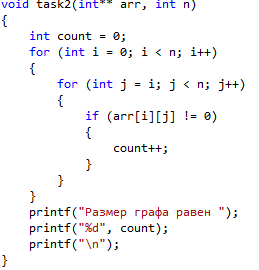
**Задание 1**

1. Сгенерируйте(используя генератор случайных чисел) матрицу смежности для неориентированного взвешенного графа G. Выведите матрицу на экран
2. Определите размер графа G, используя матрицу смежности графа.
3. Найдите изолированные, концевые и доминирующие вершины.

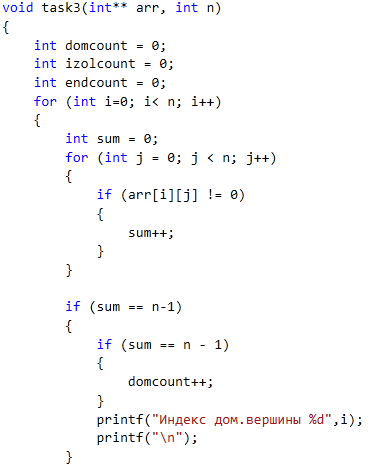
**Пояснительный текст к программе:**

****

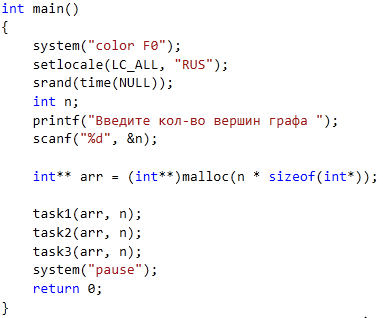
Первая функция отвечает за генерацию матрицы смежности взвешенного графа.



Во второй функции реализуется подсчет кол-ва ребер для выяснения размера графа.

****

В третьей функции реализовывается подсчет и вывод кол-ва доминантных, изолированных и концевых вершин.

****

В основном теле программы выделяется память под матрицу смежности графа через malloc, осуществляется ручной ввод пользователем кол-ва вершин графа и вызываются описанные выше функции.

**Листинг:**

**Lab5.1**

#include "stdafx.h"

#define \_CRT\_SECURE\_NO\_WARNINGS

#include <stdio.h>

#include <stdlib.h>

#include <time.h>

#include <Windows.h>

#include <iostream>

void task1(int\*\* arr, int n)

{

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

{

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

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

{

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

}

}

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

{

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

{

if (i == j)

{

arr[i][j] = 0;

}

else

arr[i][j] = arr[j][i];

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

}

printf("\n");

}

}

void task2(int\*\* arr, int n)

{

int count = 0;

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

{

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

{

if (arr[i][j] != 0)

{

count++;

}

}

}

printf("Размер графа равен ");

printf("%d", count);

printf("\n");

}

void task3(int\*\* arr, int n)

{

int domcount = 0;

int izolcount = 0;

int endcount = 0;

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

{

int sum = 0;

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

{

if (arr[i][j] != 0)

{

sum++;

}

}

if (sum == n-1)

{

if (sum == n - 1)

{

domcount++;

}

printf("Индекс дом.вершины %d",i);

printf("\n");

}

if(sum == 0)

{

if (sum == 0)

{

izolcount++;

}

printf("Индекс изол.вершины %d",i);

printf("\n");

}

if (sum == 1)

{

if (sum == 1)

{

endcount++;

}

printf("Индекс конц.вершины %d",i);

printf("\n");

}

}

printf("Кол-во доминантных вершин %d", domcount);

printf("\n");

printf("Кол-во изолированных вершин %d", izolcount);

printf("\n");

printf("Кол-во концевых вершин %d", endcount);

printf("\n");

}

int main()

{

system("color F0");

setlocale(LC\_ALL, "RUS");

srand(time(NULL));

int n;

printf("Введите кол-во вершин графа ");

scanf("%d", &n);

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

task1(arr, n);

task2(arr, n);

task3(arr, n);

system("pause");

return 0;

}

**Вывод:** Мы научились создавать графы и находить основные типы вершин в них и приобрели практические навыки решения задач с использованием основ теории графов.