МІНІСТЕРСТВО ОСВІТИ ТА НАУКИ УКРАЇНИ

Одеський національний політехничний університет

Інститут Комп'ьютерних Систем

Кафедра Інформаційних Систем

Отчет по лабораторной работе №5

«Графы: способы их хранения и обхода (в ширину и в глубину). Проверка графа на двудольность, поиск циклов и топологическая сортировка графа»

Выполнил:

студент группы АД-171

Надводський В.В.

Проверил:

Шибаева Н. О.

Одесса 2019

**Цель :** Приобретение практических навыков программирования с использованием графов. Способы представления графов и работа с графами.

**Вариант 12** В заданном графе необходимо определить, существует ли цикл, проходящий по каждому ребру графа ровно один раз.

Код

[#include](tg://search_hashtag?hashtag=include) <conio.h>  
int matrica\_smeznosti[ 50 ][ 50 ];  
int top[ 50 ];  
int inout[ 50 ][ 2 ];

int poisk\_cikla( int size )

{  
int count\_1, count\_2;  
int id = 0;  
for( count\_1 = 0; count\_1 < size; count\_1++ ) //проходим по всей матрице

{ //и считаем, сколько ребер  
for( count\_2 = 0; count\_2 < size; count\_2++ ) //входят и выходят из каждой

{ //вершины

if( matrica\_smeznosti[ count\_1 ][ count\_2 ] == 1 )  
{

inout[ count\_1 ][ 0 ] += 1; //то что получили заносим в  
} //матрицу inout, где номер

if( matrica\_smeznosti[ count\_2 ][ count\_1 ] == 1) //слолбца совпадает с номером

{ //вершины  
inout[ count\_1 ][ 1 ] += 1;  
}  
}  
}  
for( count\_1 = 0; count\_1 < size; count\_1++ ) //проходим по матрице inout  
{ //и сравниваем числа в каждом  
if( inout[ count\_1 ][ 0 ] != inout[ count\_1 ][ 1 ] ) //столбце, если найдется  
{ //хотя бы одно несовпадение  
id += 1; //то Эйлерового цикла нет  
}  
}  
if(id == 0) return 0;  
else return 1;  
}  
  
int main()  
{  
int versini, rebra;  
int count\_1, count\_2, i, j;  
printf( "VVedite kol-vo versin i reber v grafe\n" );  
scanf( "%d %d", &versini, &rebra );  
printf( "Vvodite rebra\n" );  
for( i = 0; i < rebra; i++ ) //считываем колличество ребер  
{  
scanf( "%d %d", &count\_1, &count\_2 ); //считываем ребра  
matrica\_smeznosti[ count\_1 ][ count\_2 ] = 1;  
}  
if( poisk\_cikla( versini ) == 0 ) //ищем Эйлеров путь в графе  
{  
printf( "Ailerov cikl est'\n" );  
}  
else  
{  
printf( "Cikla net\n" );  
}  
getch();  
return 0;  
}

Начало формы

Конец формы

Скрин

**Вывод:** в данной лабораторной работе я воспользовался данными с теоретической части и найденным дополнительным материалом сделал лабораторную работу а так же блок схему к выполнению процесса.