Министерство 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 «Обход графа в ширину»

Выполнил: студент группы 19ВВ2

Трошкин В.К.

Принял: Д.Т.Н., профессор

Митрохин М.А.

Пенза 2020

**Цель работы:** составить программу выполнения алгоритма обхода графа в ширину на ЭВМ.

**Ход работы:**

1. Сгенерировали (используя генератор случайных чисел) матрицу *M* смежности неориентированного помеченного графа *G*. Вывели сгенерированную матрицу на экран.
2. Для матричной формы представления графов выполнили обход в ширину.
3. Результат выполнения операции вывели на экран.

**Листинг:**

Header.h

#pragma once

int pr\_adj\_matrix(char\*\* matrix, int size);

char\*\* gen\_adj\_matrix(int size);

void POSH(char\*\* matrix, int size);

main.cpp

#define \_CRT\_SECURE\_NO\_WARNINGS

#include <stdio.h>

#include <conio.h>

#include <stdlib.h>

#include <windows.h>

#include "Header.h"

int main() {

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

int size;

printf("Введите размер матрицы: ");

scanf("%d", &size);

char\*\* adj = gen\_adj\_matrix(size);

printf("Матрица смежности: \n");

pr\_adj\_matrix(adj, size);

printf("\n");

printf("Порядок обхода: \n");

POSH(adj, size);

\_getch();

}

adjmatrixgen.cpp

#include <time.h>

#include <stdlib.h>

char\*\* gen\_adj\_matrix(int size) {

srand(time(0));

char\*\* matrix = (char\*\*)calloc(size, sizeof(char\*));

if (matrix == NULL)

return NULL;

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

matrix[i] = (char\*)calloc(size, sizeof(char));

if (matrix[i] == NULL)

return NULL;

}

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

for (int j = i; j < size; j++) {

if (i == j) {

matrix[i][j] = 0;

}

else {

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

matrix[j][i] = matrix[i][j];

}

}

}

return matrix;

}

BFS.cpp

#include <stdlib.h>

#include <stdio.h>

#include <queue>

using namespace std;

int find\_unvisited(char\* visited, int size) {

int unvisited = -1;

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

if (!visited[i]) {

unvisited = i;

break;

}

}

return unvisited;

}

void BFS(char\*\* matrix, char\* visited, int size, int v) {

queue <int> q;

q.push(v);

visited[v] = 1;

while (!q.empty()) {

v = q.front();

q.pop();

printf("%d ", v);

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

if ((matrix[v][i]) && (!visited[i])) {

q.push(i);

visited[i] = 1;

}

}

}

}

void POSH(char\*\* matrix, int size) {

int counter = 0;

char\* visited = (char\*)calloc(size, sizeof(char));

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

visited[i] = 0;

}

int v = find\_unvisited(visited, size);

while (v != -1) {

BFS(matrix, visited, size, v);

counter++;

v = find\_unvisited(visited, size);

printf("\n");

}

printf("\nЧисло компонент связности: \n");

printf("%d", counter);

}

pr\_output.cpp

#include <stdio.h>

int pr\_adj\_matrix(char\*\* matrix, int size) {

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

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

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

}

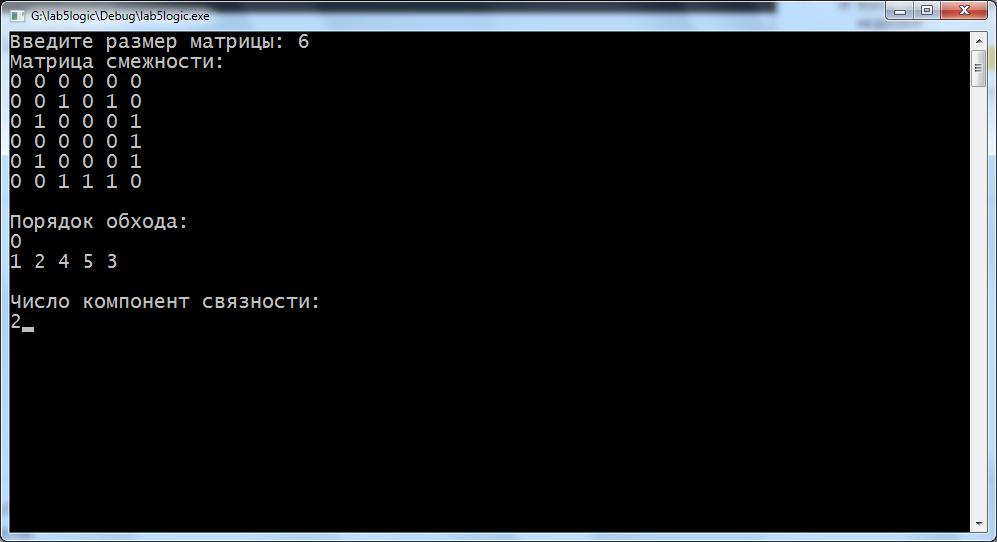
printf("\n");

}

return 0;

}

Результаты работы программы:



**Вывод:** составили программу обхода графа в ширину на ЭВМ.