|  |
| --- |
| Министерство образования Российской Федерации  Пензенский государственный университет  Кафедра «Вычислительная техника» |
| Отчет  по лабораторной работе №8  по курсу «Логика и основы алгоритмизации в инженерных задачах»  на тему «Определение характеристик графов» |
|  |
|  |
| Выполнила студент группы 19ВВ2:  Отставнов А.М.  Принял:  Митрохин М. А. |
| Пенза  2020 |

**Задание 1**

1. Сгенерируйте (используя генератор случайных чисел) матрицу

смежности для неориентированного взвешенного графа G. Выведите матрицу

на экран.

2. Определите радиус и диаметр графа G, используя матрицу смежности

графа.

3. Определите подмножества периферийных и центральных вершин

графа G, используя матрицу смежности.

4. Найдите изолированные, концевые и доминирующие вершины.

**Листинг**

#include <stdio.h>

#include <conio.h>

#include <Windows.h>

#include <stdlib.h>

#include <queue>

#include <ctime>

#include <iostream>

using namespace std;

int N, i, j, C=0;

int \*\*graph, \*dist, \*ex;

int BFS(int \*\*graph, int \*dist, int i, int N) {

queue <int> Q;

int j;

Q.push(i);

dist[i] = 0;

while (Q.empty() != true) {

i = Q.front();

Q.pop();

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

if ((graph[i][j] != 0) && (dist[j] > dist[i] + graph[i][j])) {

dist[j] = 0;

Q.push(j);

dist[j] = dist[i] + graph[i][j];

}

}

}

return 0;

}

int main(void)

{

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

printf("Введите количество вершин - ");

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

srand(time(NULL));

dist = new int[N];

ex = new int[N];

graph = new int\*[N];

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

graph[i] = new int[N];

}

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

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

graph[i][j] = rand() % 5;

graph[j][i] = graph[i][j];

graph[i][i] = graph[j][j] = 0;

}

}

printf("\n ");

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

{

printf("%4d ", j + 1); //горизонт

}

printf("\n\n");

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

{

printf(" %d ", i + 1); // вертик

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

{

printf("%4d ", graph[i][j]);

}

printf("\n\n");

}

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

dist[i] = 1000;

}

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

ex[i] = -1;

}

int D = -1, r = 10000;

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

BFS(graph, dist, i, N);

int k;

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

if ((ex[i] < dist[k]) && (dist[k]!=1000)) {

ex[i] = dist[k];

}

}

if ((ex[i] < r)&&(ex[i] != 0 )) {

r = ex[i];

}

if (ex[i] > D) {

D = ex[i];

}

for (int l = 0; l < N; l++) {

dist[l] = 1000;

}

}

printf(" Вершина : ");

for (int i = 1; i <= N; i++) {

printf(" %d |", i);

}

printf("\n Эксцентириситет : ");

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

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

}

printf("\n\n Радиус графа : %d\n", r);

printf(" Диаметр графа : %d\n", D);

printf(" Центральные вершины :");

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

if (ex[i] == r) {

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

}

}

printf("\n Периферийные вершины :");

for (int n = 0; n < N; n++) {

if (ex[n] == D) {

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

}

}

printf("\n Изолированные вершины :");

for (int n = 0; n < N; n++) {

if (ex[n] == 0) {

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

}

}

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

int rank = 0;

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

if (graph[i][j] != 0) {

rank++;

}

}

if (rank == 1) {

printf("\n Вершина %d концевая", i);

}

if (rank == N-1) {

printf("\n Вершина %d доминирующая", i+1);

}

}

free(graph);

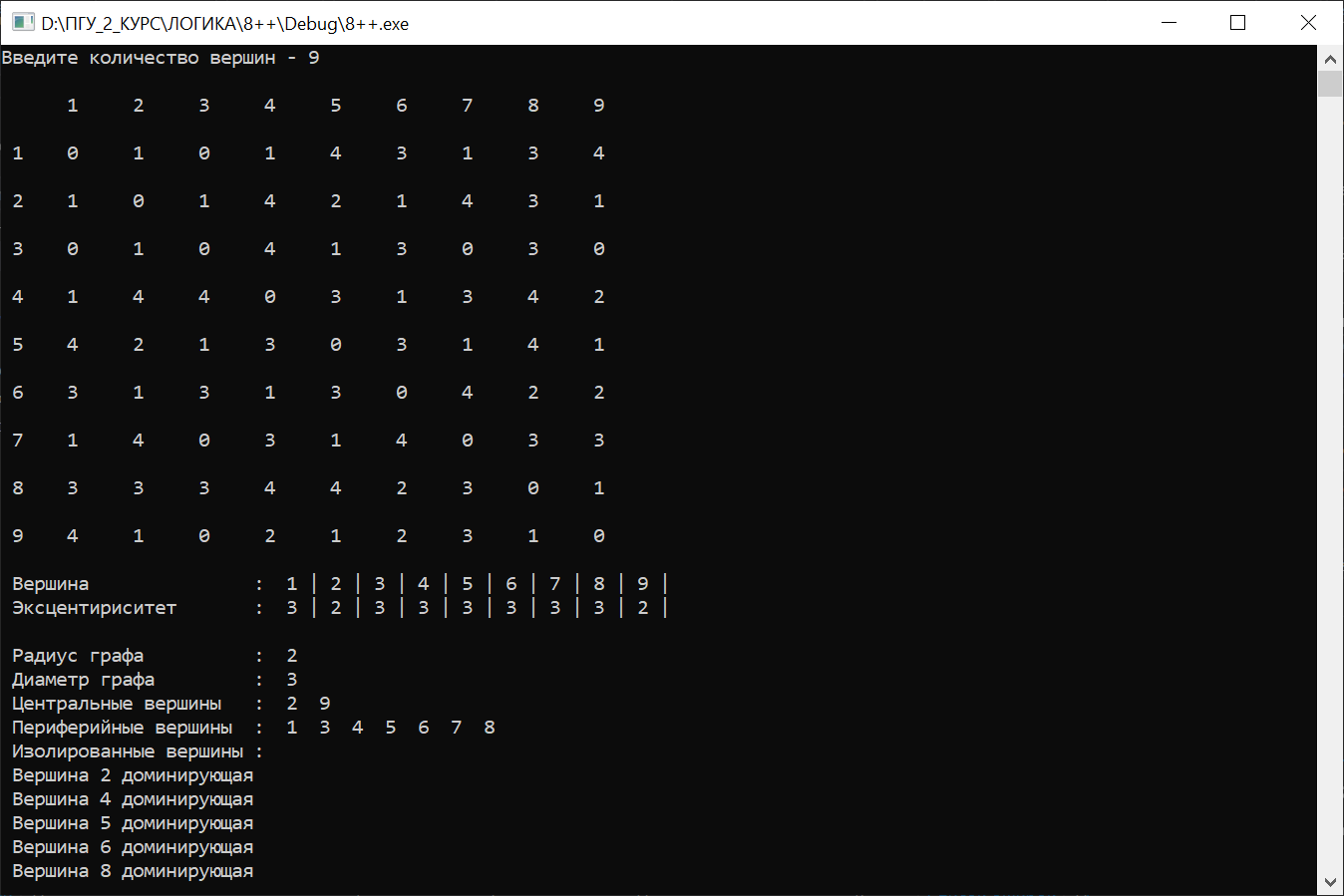
free(dist);

free(ex);

\_getch();

}

**Результат работы программы:**



**Вывод:**

В результате работы программы на матрице смежности графа были найдены эксцентриситеты вершин, радиус и диаметр графа, множества центральных, периферийных, изолированных, концевых и доминирующих вершин.