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»

**Выполнил:**

студент группы 20ВВ3

Баулин А.Д.

**Приняли:**

Юрова О.В.

Митрохин М.А.

Пенза 2021

# Название

# определение характеристик графов.

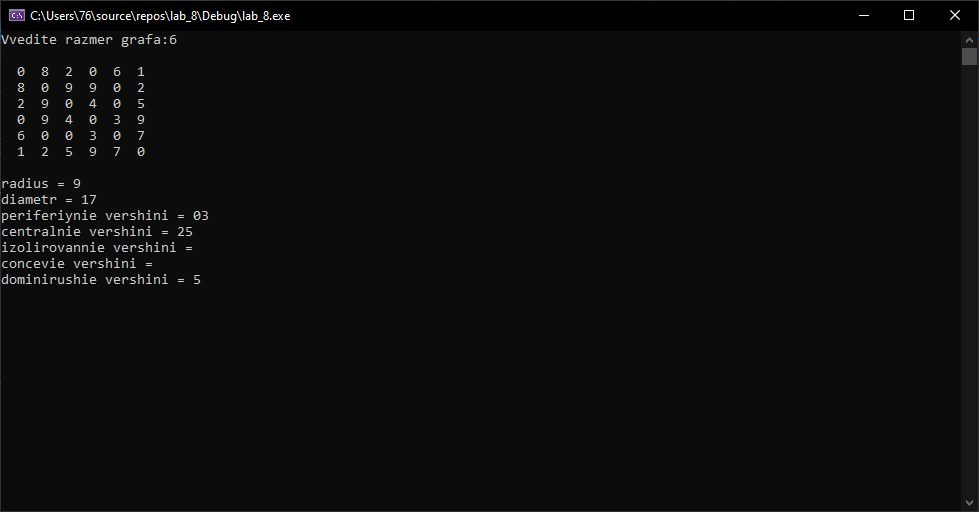
# Лабораторное задание

**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.

# Результат работы программы

**Задание 1.1.**Результаты работы программы показаны на рисунке 1.



**Рисунок 1 – Результат работы программы**

**Вывод:** изучил алгоритмы для определения характеристик графа.

**Листинг**

#define \_CRT\_SECURE\_NO\_WARNINGS

#include <conio.h>

#include <stdlib.h>

#include <stdio.h>

#include <time.h>

#include <stdbool.h>

#include <queue>

#include <string.h>

using namespace std;

void print(int\*\* M, int n)

{

int i, j;

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

{

printf("\n");

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

{

printf("%\*d", 3, M[i][j]);

}

}

}

int\*\* gen\_s(int n)

{

int i, j, z = 1;

int\*\* M;

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

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

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

//srand(time(NULL));

srand(12);

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

{

M[i][i] = 0;

for (j = z; j < n; j++)

{

if (rand() % 10 > 0) M[i][j] = rand() % 10;

else M[i][j] = 0;

M[j][i] = M[i][j];

}

z++;

}

return M;

}

void inf\_1(int\*\* M, int n)

{

int i, j, r, d;

int\* m, \* v;

queue <int> Q;

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

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

{

m[i] = -1;

}

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

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

{

Q.push(z);

m[z] = 0;

while (!Q.empty())

{

j = Q.front();

Q.pop();

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

{

if (M[j][i] > 0 && m[i] == -1)

{

m[i] = m[j] + M[j][i];

Q.push(i);

}

}

}

v[z] = 0;

d = 0;

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

{

if (m[i] > d)

{

v[z] = m[i];

d = m[i];

}

}

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

{

m[i] = -1;

}

}

d = 0;

r = INT\_MAX;

for (i = 0; i < n - 1; i++)

{

if (v[i] > d)

d = v[i];

if (v[i] < r)

r = v[i];

}

printf("\n\nradius = %d", r);

printf("\ndiametr = %d", d);

printf("\nperiferiynie vershini = ");

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

{

if (v[i] == d)

printf("%d", i);

}

printf("\ncentralnie vershini = ");

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

{

if (v[i] == r)

printf("%d", i);

}

printf("\nizolirovannie vershini = ");

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

{

d = 0;

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

{

if (M[i][j] > 0) d++;

}

if (d == 0)

printf("%d", i);

}

printf("\nconcevie vershini = ");

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

{

d = 0;

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

{

if (M[i][j] > 0) d++;

}

if (d == 1)

printf("%d", i);

}

printf("\ndominirushie vershini = ");

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

{

d = 0;

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

{

if (M[i][j] > 0) d++;

}

if (d == n - 1)

printf("%d", i);

}

}

void main()

{

int\*\* M = NULL, n, x = 0;

printf("Vvedite razmer grafa:");

scanf("%d", &n);

M = gen\_s(n);

print(M, n);

inf\_1(M, n);

\_getch();

}