Пензенский государственный университет

Кафедра «Вычислительная техника»

**ОТЧЕТ**

по лабораторной работе №7

# по курсу «Логика и основы алгоритмизации в инженерных задачах»

на тему «поиск расстояния во взвешенном графе»

**Выполнил:**

студент группы 20ВВ3

Баулин А.Д.

**Приняли:**

Юрова О.В.

Митрохин М.А.

Пенза 2021

# Название

Поиск расстояний во взвешенном графе.

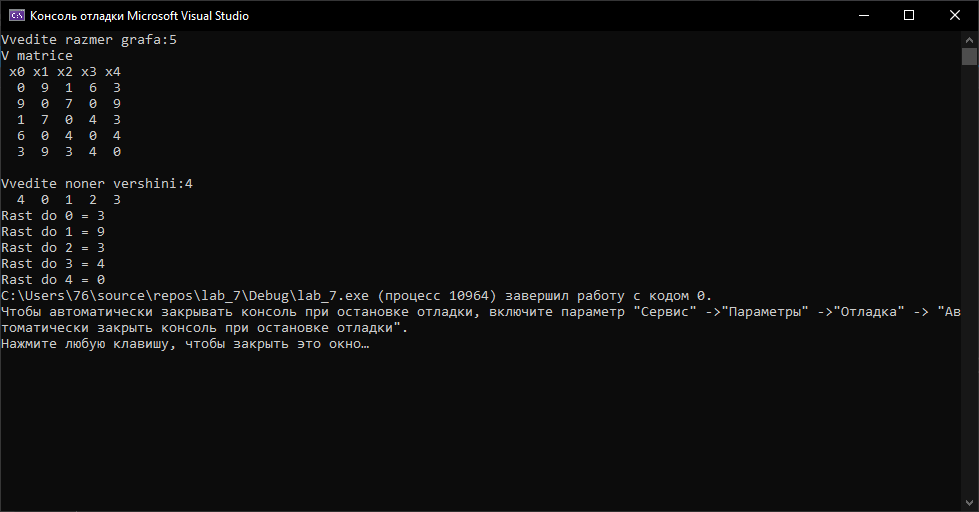
# Лабораторное задание

**Задание 1**

1. Сгенерируйте (используя генератор случайных чисел) матрицу смежности для неориентированного взвешенного графа *G*. Выведите матрицу на экран.
2. Для сгенерированного графа осуществите процедуру поиска расстояний, реализованную в соответствии с приведенным выше описанием. При реализации алгоритма в качестве очереди используйте класс **queue**из стандартной библиотеки С++.

# Результат работы программы

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



**Рисунок 1 – Результат работы программы**

**Вывод:** изучил алгоритм поиска расстояний во взвешенном графе.

**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;

int\*\* gen\_m\_v(int\*\* M, int n)

{

int i, j, z = 1;

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

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

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

srand(time(NULL));

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

{

M[i][i] = 0;

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

{

if (rand() % 100 > 50) M[i][j] = 0; else M[i][j] = 1;

if (M[i][j] = 1)

{

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

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

}

}

z++;

}

return M;

}

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

{

int i, j;

printf("V matrice\n");

for (i = 0; i < n; i++) printf(" x%d", i);

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

{

printf("\n");

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

{

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

}

}

}

void ob\_m\_1(int\*\* M, int n, int k, int\* m)

{

queue <int> Q;

int i, j = k;

Q.push(k);

m[k] = 0;

while (!Q.empty())

{

j = Q.front();

printf("%\*d", 3, j);

Q.pop();

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

{

if (M[j][i] > 0 && m[i] > m[j] + M[j][i])

{

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

Q.push(i);

}

}

}

}

void hod1(int\*\* M, int n, int k)

{

int i, \* m;

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

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

{

m[i] = INT\_MAX;

}

ob\_m\_1(M, n, k, m);

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

if (m[i] == INT\_MAX)

printf("\nRast do %d net", i);

else

printf("\nRast do %d = %d", i, m[i]);

free(m);

}

void main()

{

int\*\* M = NULL, n, k, i;

printf("Vvedite razmer grafa:");

scanf("%d", &n);

M = gen\_m\_v(M, n);

print(M, n);

printf("\n\nVvedite noner vershini:");

scanf("%d", &k);

hod1(M, n, k);

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

{

free(M[i]);

}

free(M);

}