МИНОБРНАУКИ РОССИИ

Санкт-Петербургский государственный

электротехнический университет

«ЛЭТИ» им. В.И. Ульянова (Ленина)

Кафедра САПР

отчет

авпывпав

**по курсовапой работе за 1 семестр**

тема: «**обработка и сортировка элементов квадратной матрицы с использованием спирального обхода»**

вариант: 18

|  |  |  |
| --- | --- | --- |
| Студент гр. 4351 |  | Чмырева В.Д. |
| Преподаватель |  | Рыжов Н.Г. |

Санкт-Петербург

2024

**Цель работы.**

Продемонстрировать навыки работы с двумерными массивами (матрицами) в языке программирования C++

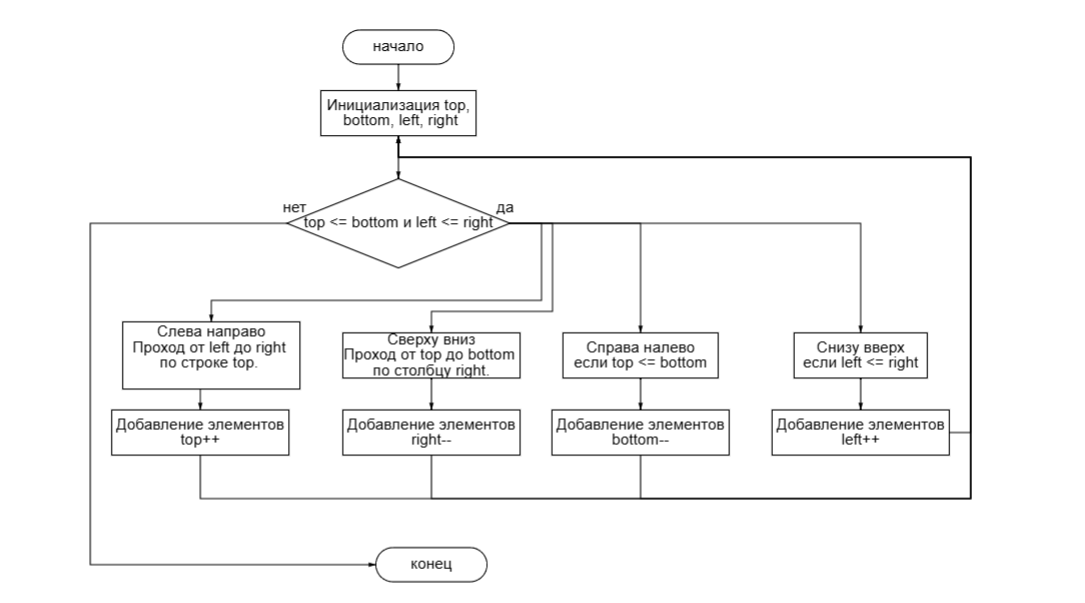
**Задание.**

Дана целочисленная квадратная матрица размерности n. Расположить все элементы матрицы по убыванию. Обход матрицы осуществлять по спирали по часовой стрелке от левого верхнего угла к центру. Размерность n и исходная матрица вводятся из текстового файла.

**Схема алгоритма работы программы.**



**Обход матрицы по спирали**



**Код программы.**

#include <iostream>

#include <fstream>

#include <vector>

#include <algorithm>

#include <locale>

using namespace std;

void printMatrix(const vector<vector<int>>& matrix) {

for (const auto& row : matrix) {

for (int val : row) {

cout << val << " ";

}

cout << endl;

}

}

int main() {

setlocale(LC\_ALL, "rus");

ifstream inputFile("matrix.txt");

if (!inputFile) {

cerr << "Ошибка открытия файла." << endl;

return 1;

}

int n;

inputFile >> n;

vector<vector<int>> matrix(n, vector<int>(n));

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

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

inputFile >> matrix[i][j];

}

}

inputFile.close();

// Обход матрицы по спирали и сохранение элементов в вектор

vector<int> spiralElements;

int top = 0, bottom = n - 1, left = 0, right = n - 1;

while (top <= bottom && left <= right) {

// Слева направо

for (int i = left; i <= right; ++i) {

spiralElements.push\_back(matrix[top][i]);

}

top++;

// Сверху вниз

for (int i = top; i <= bottom; ++i) {

spiralElements.push\_back(matrix[i][right]);

}

right--;

// Справа налево

if (top <= bottom) {

for (int i = right; i >= left; --i) {

spiralElements.push\_back(matrix[bottom][i]);

}

bottom--;

}

// Снизу вверх

if (left <= right) {

for (int i = bottom; i >= top; --i) {

spiralElements.push\_back(matrix[i][left]);

}

left++;

}

}

// Сортировка элементов векторе по убыванию

sort(spiralElements.begin(), spiralElements.end(), greater<int>());

// Заполнение матрицы отсортированными элементами по спирали

int index = 0;

top = 0, bottom = n - 1, left = 0, right = n - 1;

while (top <= bottom && left <= right) {

// Слева направо

for (int i = left; i <= right; ++i) {

matrix[top][i] = spiralElements[index++];

}

top++;

// Сверху вниз

for (int i = top; i <= bottom; ++i) {

matrix[i][right] = spiralElements[index++];

}

right--;

// Справа налево

if (top <= bottom) {

for (int i = right; i >= left; --i) {

matrix[bottom][i] = spiralElements[index++];

}

bottom--;

}

// Снизу вверх

if (left <= right) {

for (int i = bottom; i >= top; --i) {

matrix[i][left] = spiralElements[index++];

}

left++;

}

}

cout << "Отсортированная матрица по спирали:" << endl;

printMatrix(matrix);

return 0;

}

**Описание программы.**

Программа считывает размерность и элементы матрицы с текстового файла matrix.txt. Далее создает двумерный вектор размерностью матрицы и после чего он заполняется элементами матрицы. После чего элементы сортируются по убыванию. В конечном итоге все элементы заполняются в матрицу и выводятся на экран.

**Руководство пользователя.**

Убедитесь, что рядом с исполняемым файлом находится файл matrix.txt с размерностью матрицей(1) и ее элементами(2):

• (1): n (должно быть больше 1).

• (2): matrix (элементы матрицы).

Запустите программу:

На экран выведется отсортированная матрица.

**Пути дальнейшего улучшения.**

Оптимизировать алгоритм сортировки, добавить более эффективный для быстрой работы с большой размерностью матрицы. Добавить удобный интерфейс.

**Вывод.**

В данной работе была разработана программа, которая демонстрирует, как работать с двумерными массивами (матрицами), позволяя нам понять, как организовать данные в виде строк и столбцов при помощи алгоритма сортировок.