Министерство науки и высшего образования Российской Федерации Федеральное государственное автономное образовательное учреждение высшего образования

**«Пермский национальный исследовательский политехнический университет»**

Электротехнический факультет

Кафедра «Информационные технологии и автоматизированные системы»

Направление подготовки: 09.03.01 – «Информатика и вычислительная техника»

**Лабораторная работа № 5**

по дисциплине

**«Основы алгоритмизации и программирования»**

Выполнил студент гр. ИВТ-21-1б

Мифтахов Ильяс Гарифжанович

Проверил:

ст. преп. каф. ИТАС

Яруллин Денис Владимирович

(оценка) (подпись)

(дата)

**Постановка задачи**

Используя функции, решить указанную в варианте задачу. Массив должен передаваться в функцию как параметр.

**Задание**

Задана строка из N2 цифр. Установить можно ли, разбив строку на подстроки длиной N, записать их в строки двумерного массива N x N по одной цифре в одном элементе так, чтобы они в первом столбце расположились в порядке возрастания.

**Основная часть**

Текст программы представлен ниже:

#include <iostream>

#include <string.h>

#include <cmath>

#include <pch.h>

using namespace std;

void show\_matrix(char \*\* a, int size)

{

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

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

cout.width(3);

cout << a[i][j];

}

cout << endl;

}

cout << endl;

}

char\*\* sort\_first\_column(char \*\* matrix, int size)

{

for (int i = 0; i < size - 1; i++)

{

for (int j = i + 1; j < size; j++)

{

if (matrix[i][0] > matrix[j][0])

{

for (int m = 0; m < size; m++)

{

char c = matrix[i][m];

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

matrix[j][m] = c;

}

}

}

}

return matrix;

}

bool are\_equal\_matrixes(char \*\* old\_matrix, char \*\* new\_matrix, int matrix\_size)

{

for (int i = 0; i < matrix\_size; i++)

{

for (int j = 0; j < matrix\_size; j++)

{

for (int m = 0; m < matrix\_size; m++)

{

if (old\_matrix[i][0] != new\_matrix[j][0])

{

return false;

}

}

}

}

return true;

}

int main()

{

string s;

int n, k = 0;

cin >> n;

int matrix\_size = sqrt(n);

char\*\* old\_matrix = new char\*[matrix\_size];

cin >> s;

for (int i = 0; i < matrix\_size; i++)

old\_matrix[i] = new char[matrix\_size];

for (int i = 0; i < matrix\_size; i++)

{

for (int j = 0; j < matrix\_size; j++)

{

old\_matrix[i][j] = s.at(k);

k++;

}

}

cout << endl << "Old Matrix" << endl;

show\_matrix(old\_matrix, matrix\_size);

char\*\* new\_matrix = sort\_first\_column(old\_matrix, matrix\_size);

cout << "New Matrix" << endl;

show\_matrix(new\_matrix, matrix\_size);

if (!are\_equal\_matrixes(old\_matrix, new\_matrix, matrix\_size))

{

cout << "Can" << endl;

}

else

{

cout << "Can't" << endl;

}

delete new\_matrix, old\_matrix;

return 0;

}

Для корректной работы программы со строкой были подключены дополнительные директивы препроцессора – <string.h>, <cmath>, <pch.h>

Итак, для начала были подключены стандартные директивы препроцессора, указано пространство «std». Далее, в главной функции мы работаем с конечными матрицами, сравнивая их, вызываем подфункции.

В первой функции заполняем двумерный массив(создаем матрицу). Во второй функции производим упорядочивание первого столбца матрицы. В третьей функции проверяем правильность выполнения остальных функций и действий.

На рисунках 1-4 представлена блок-схема программы.



Рисунок 1

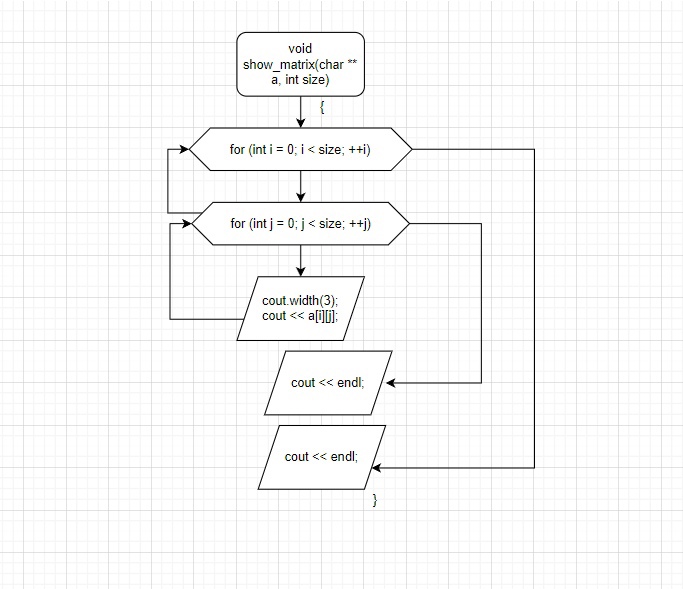


Рисунок 2

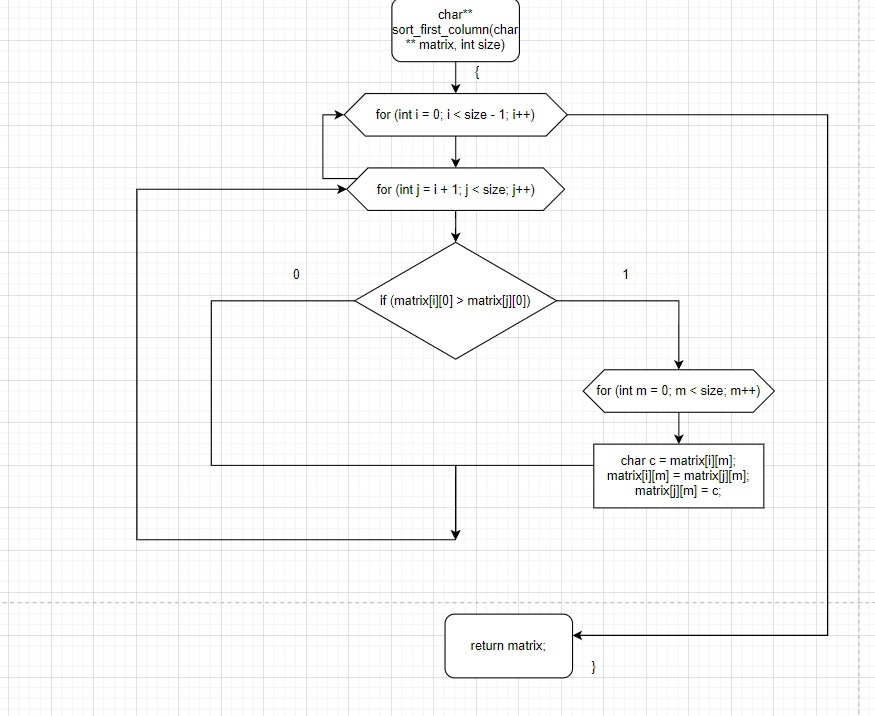


Рисунок 3

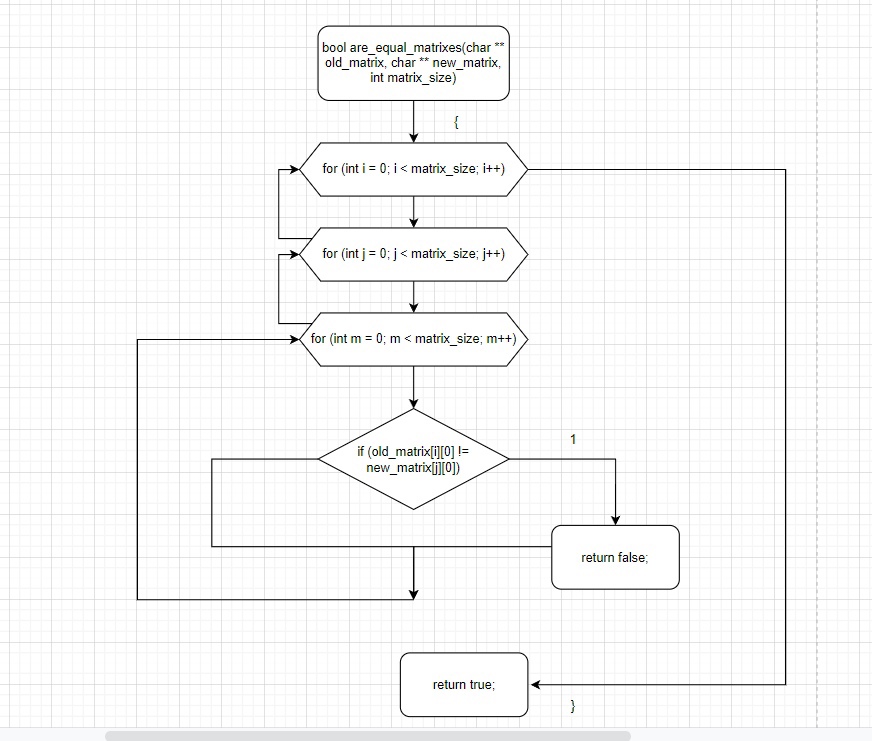


Рисунок 4

На рисунке 5 представлен результат выполнения программы.

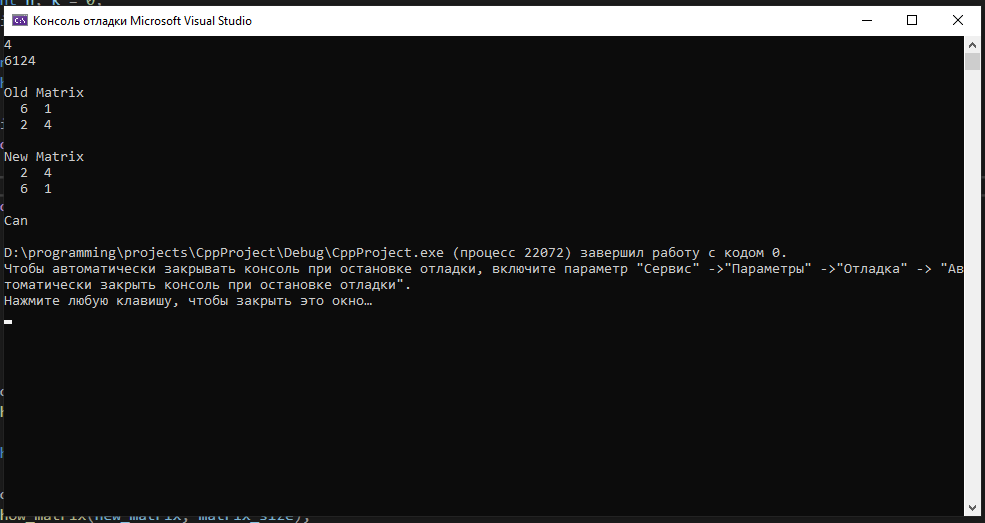


Рисунок 5

**Вывод**

В лабораторной работе №5 поставленная задача выполнена, использованы подфункции, динамические массивы, работа со строками. На экран выводятся корректные результаты, функции работают исправно.