|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  |  | | --- | --- | --- | |  |  |  | | МИНОБРНАУКИ РОССИИ | | | | Федеральное государственное бюджетное образовательное учреждение  высшего образования  **«МИРЭА** – **Российский технологический университет»**  **РТУ МИРЭА** | | |   Институт информационных технологий |
| Кафедра вычислительной техники |
|  |

|  |  |
| --- | --- |
| **ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ №2** | |
| **по дисциплине** | |
| *«Структуры и алгоритмы обработки данных»* | |
| Выполнил студент группы *ИКБО-02-17* | *Данилов Д.Е.* |
| Принял | *Асадова Ю.С.* |

|  |  |  |
| --- | --- | --- |
| Лабораторные работы выполнены | «\_\_»\_\_\_\_\_\_\_201\_\_ г. | *\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_*  *Подпись студента* |
|  |  |  |
| «Зачтено» | «\_\_»\_\_\_\_\_\_\_201\_\_ г. | *\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_*  *Подпись руководителя* |

Москва 2018

|  |  |  |
| --- | --- | --- |
|  |  |  |
| МИНОБРНАУКИ РОССИИ | | |
| Федеральное государственное бюджетное образовательное учреждение  высшего образования  **«МИРЭА** – **Российский технологический университет»**  **РТУ МИРЭА** | | |

Институт информационных технологий

Кафедра вычислительной техники

Выполнено \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/Д.Е. Данилов/

Зачтено \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/Ю.С. Асадова/

**Задание на лабораторную работу**

**Дисциплина**: **«**Структуры и алгоритмы обработки данных**»**

**Студент** Данилов Д.Е. **Шифр** 17и0281 **Группа** ИКБО-02-17

**1. Тема**: «Перегрузка функций»

**2. Срок сдачи студентом законченной работы:** \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_г.

**3. Исходные данные:** индивидуальная тема, среда разработки, средства языка С++

**4. Задание:** Разработать программу, которая вводит матрицу из n строк и m столбцов (n<=100, m<=50) и упорядочивает элементы матрицы.

**5. Содержание отчета:**

* титульный лист;
* задание;
* оглавление;
* введение;
* основные разделы отчета;
* заключение;

Руководитель работы Ю.С. Асадова \_\_\_\_\_\_\_\_\_\_\_\_ «\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_ 2018г.

подпись

Задание принял к исполнению И.О. Фамилия \_\_\_\_\_\_\_\_\_\_\_\_ «\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_ 2018г.

подпись

*Оглавление*

[Введение 4](#_Toc524637062)

[Основные разделы отчёта 5](#_Toc524637063)

Заключение …………………………………………………………………………. 9

# *Введение*

Целью данной лабораторной работы является закрепление знаний, полученных при изучении языка С++, а именно: работа по теме перегрузка функций. Лабораторная работа была выполнена в среде разработки Microsoft Visual Studio 2017 на языке C++.

1. ***Теоретическое введение***

C++ позволяет определять несколько функций с одинаковым именем в одной области. Такие функции называются перегруженными и подробно описываются в разделе "Перегрузка". Перегруженные функции позволяют программистам указывать для функций разную семантику в зависимости от типов и числа аргументов.

Например, функция **print**, принимающая строковый аргумент (или **char \***), выполняет задачи, совершенно отличающиеся от задач, выполняемых функцией, принимающей аргумент типа **double**. Перегрузка позволяет использовать универсальные имена и препятствует применению программистами таких имен, как print\_sz или print\_d. В следующей таблице указаны компоненты объявления функций, используемые языком C++ для различения групп функций с одинаковым именем в одной области.

1. ***Постановка задачи***

Разработать программу, которая вводит матрицу из n строк и m столбцов (n<=100, m<=50) и упорядочивает элементы матрицы.

**Дано:** количество элементов в матрице, сами элементы.

**Ограничения на данные:** количество элементов должно быть целым числом или символом

**Ограничения на решаемую задачу**: нет

1. ***Определение входных и выходных данных.***

**Входные данные** – числа, вводится с клавиатуры или рандомно.

**Результат** – упорядоченная новая матрица

1. ***Краткое описание программы.***

Рандомно заполняется матрица. Происходит поиск сортировки. Вывод новой матрицы на экран.

**V. *Разработка программы (кодирование алгоритма)***

#include "stdafx.h"

#include <iostream>

#include <vector>

#include <ctime>

using namespace std;

void fill(vector <vector <int>> &arr, int N, int M)

{

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

arr.push\_back(vector<int>());

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

arr[i].push\_back(rand() % 100 + 1);

}

}

cout << endl;

}

void fill(vector <vector <char>> &arr, int N, int M)

{

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

arr.push\_back(vector<char>());

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

arr[i].push\_back(arr[i][j] = 'a' + rand() % ('z' - 'a'));

}

}

}

template <typename T>

void display(vector <vector <T>> arr, int N, int M)

{

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

if (i == 0) cout << endl;

else cout << endl << endl;

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

cout << arr[i][j] << '\t';

}

}

cout << endl << endl;

}

template <typename T>

void sort(vector <vector <T>> &arr, int N, int M)

{

T temp;

//Horizontal

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

for (int k = 0; k < M - 1; k++) {

for (int j = 0; j < M - k - 1; j++) {

if (arr[i][j] > arr[i][j + 1]) {

temp = arr[i][j];

arr[i][j] = arr[i][j + 1];

arr[i][j + 1] = temp;

}

}

}

}

//Vertical

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

for (int k = 0; k < N - 1; k++) {

for (int i = 0; i < N - k - 1; i++) {

if (arr[i][j] < arr[i + 1][j]) {

temp = arr[i][j];

arr[i][j] = arr[i + 1][j];

arr[i + 1][j] = temp;

}

}

}

}

}

template <typename T>

void logic(vector <vector <T>> &arr, int N, int M) {

fill(arr, N, M);

cout << "it was:" << endl;

display(arr, N, M);

sort(arr, N, M);

cout << "it is:" << endl;

display(arr, N, M);

}

void menu(short &n, int &N, int &M)

{

//setlocale(LC\_ALL, "rus");

srand(time(NULL));

cout << "int/char?" << endl;

cin >> n;

while (n < 1 || n > 2) {

cout << "another one!" << endl;

cout << "char/int?" << endl;

cin >> n;

}

cout << "Enter N & M" << endl;

cin >> N >> M;

while (N > 100 || N <= 0 || M > 50 || M <= 0) {

cout << "another one!" << endl;

cout << "Enter N & M" << endl;

cin >> N >> M;

}

}

int main()

{

short n;

int N, M;

menu(n, N, M);

switch (n)

{

case 1: {

vector <vector <int>> arr;

logic(arr, N, M);

break;

}

case 2: {

vector <vector <char>> arr(N, vector <char>(M));

logic(arr, N, M);

break;

}

default:

break;

}

system("pause");

return 0;

}

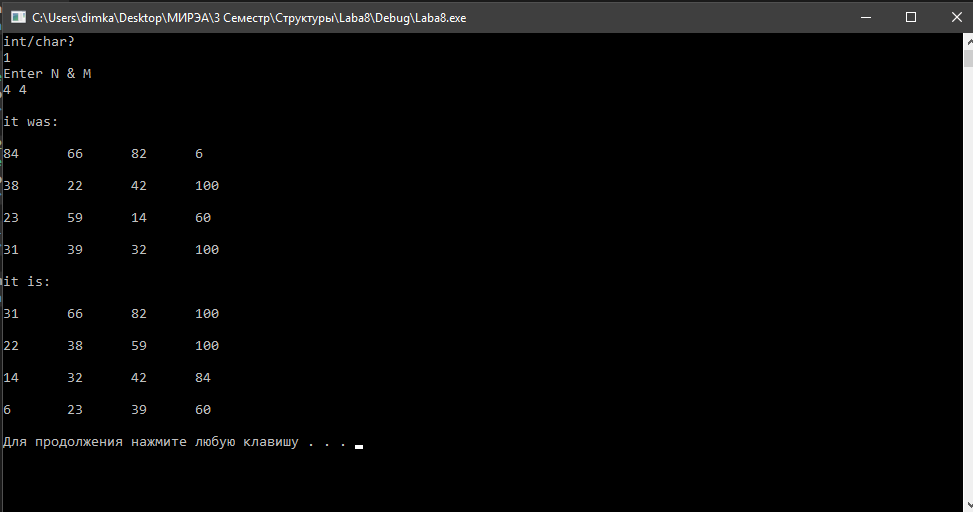


Рис.1 Результат выполнения теста 1

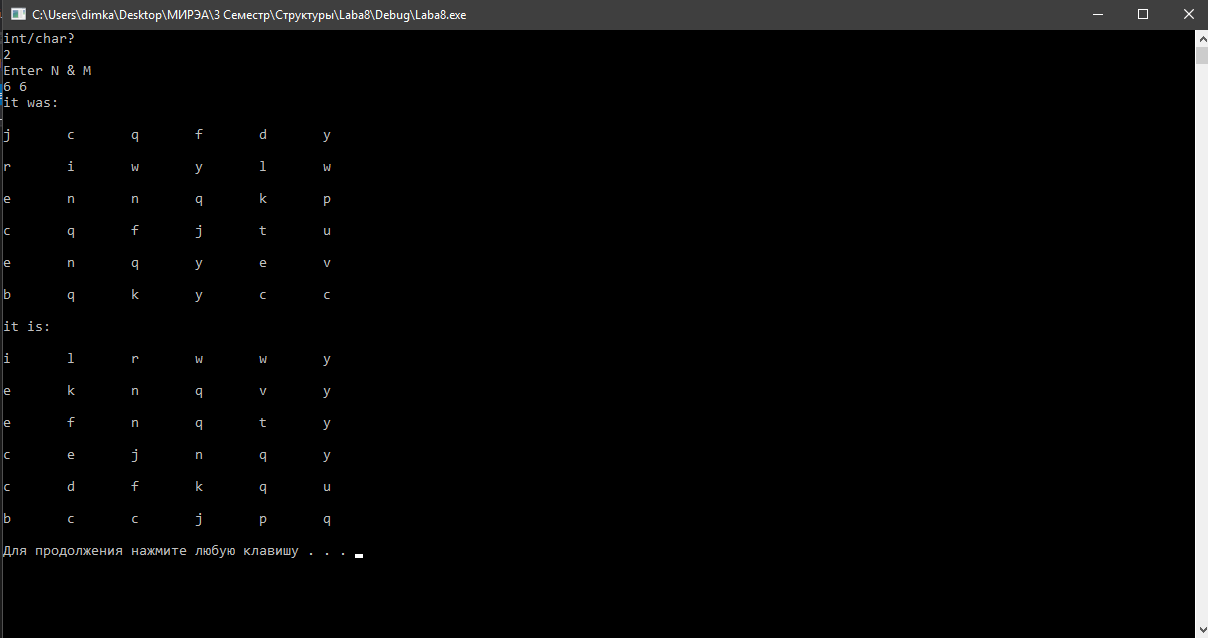


Рис.2 Результат выполнения теста 2

***Заключение***

В ходе выполнения работы были получены практические навыки при работе по теме двумерный массив и перегрузка функция на языке С++.