



МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение  
высшего образования

«МИРЭА – Российский технологический университет»

**РТУ МИРЭА**

Институт информационных технологий

Кафедра вычислительной техники

## **ОТЧЕТ ПО ПРАКТИЧЕСКОЙ РАБОТЕ №8**

**по дисциплине**

**«Алгоритмические основы обработки данных»**

Выполнил студент группы ИВБО-01-20

Д.А. Манохин

Принял старший преподаватель

Ю.С. Асадова

Практические работы выполнены

«\_\_»\_\_\_\_\_2021г.

«Зачтено»

«\_\_»\_\_\_\_\_2021г.

Москва 2021



МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение  
высшего образования

«МИРЭА – Российский технологический университет»

**РТУ МИРЭА**

Институт информационных технологий

Кафедра вычислительной техники

Выполнено \_\_\_\_\_ /Д.А. Манохин/

Зачтено \_\_\_\_\_ /Ю.С. Асадова/

## **Задание на практическую работу №8**

Дисциплина: «Алгоритмические основы обработки данных»

Студент Манохин Дмитрий Александрович Шифр 20И2132 Группа ИББО-01-20

**1. Тема:** «Перегрузка функций».

**2. Срок сдачи студентом законченной работы:** 21.10.2021г.

**3. Исходные данные:**

Размер матрицы из  $n$  строк и  $m$  столбцов ( $n \leq 100$ ,  $m \leq 50$ ).

**4. Задание:**

Разработать программу, которая вводит матрицу и упорядочивает элементы матрицы таким образом, чтобы при чтении матрицы по строкам ее элементы образовывали отсортированный по убыванию массив. Программа должна предоставлять пользователю выбор типа элементов матрицы: целого или строкового. Операции ввода, вывода и упорядочивания элементов матрицы должны быть реализованы в виде перегруженных функций.

**5. Содержание отчета:**

- титульный лист;
- задание;
- оглавление;
- введение;
- основные разделы отчета;
- заключение;
- список использованных источников;

Руководитель работы

Ю.С. Асадова

\_\_\_\_\_ «\_\_» \_\_\_\_\_ 2021г.  
подпись

Задание принял к исполнению

Д.А. Манохин

\_\_\_\_\_ «\_\_» \_\_\_\_\_ 2021г.  
подпись

# ОГЛАВЛЕНИЕ

Введение.....	4
Основной раздел.....	5
Заключение .....	13
Список использованных источников .....	14

## **ВВЕДЕНИЕ**

В данной практической работе требуется отсортировать двумерный массив, используя перегруженные функции.

Постановка задачи:

Разработать программу, запрашивающую ввод типа и размера исходного двумерного массива.

Необходима реализация проверки ввода на соответствие исходным условиям (для выбора доступно только два типа, кол-во строк двумерного массива не может быть больше 100, кол-во столбцов не может быть больше 50).

При соответствии ввода исходным условиям, будет выведено соответствующее сообщение об успехе, в противном случае – об ошибке.

Затем пользователю будет предложено ввести исходную матрицу. После ввода, матрица будет отсортирована и затем еще раз выведена в консоль. После программа завершит свою работу.

## ОСНОВНОЙ РАЗДЕЛ

В начале работы программы пользователь вводит количество строк и столбцов, если входные данные противоречат исходным условиям, то будет выведена ошибка и произведен досрочный выход из программы.

Если данные были введены правильно, то будет предложено пользователю выбор работы программы с целочисленной матрицей или же с строковой.

Далее в зависимости от выбора пользователя будет происходить работа программы, а точнее ввод, отсортировка и вывод целочисленного или строкового массива, с помощью перегруженных функций.

После работы программы и вывода результатов ее работы будет выполнен выход из нее.

Блок – схема алгоритма представлена на рисунках [1.1](#), [1.2](#), [1.3](#), [1.4](#), [1.5](#), [1.6](#), [1.7](#).

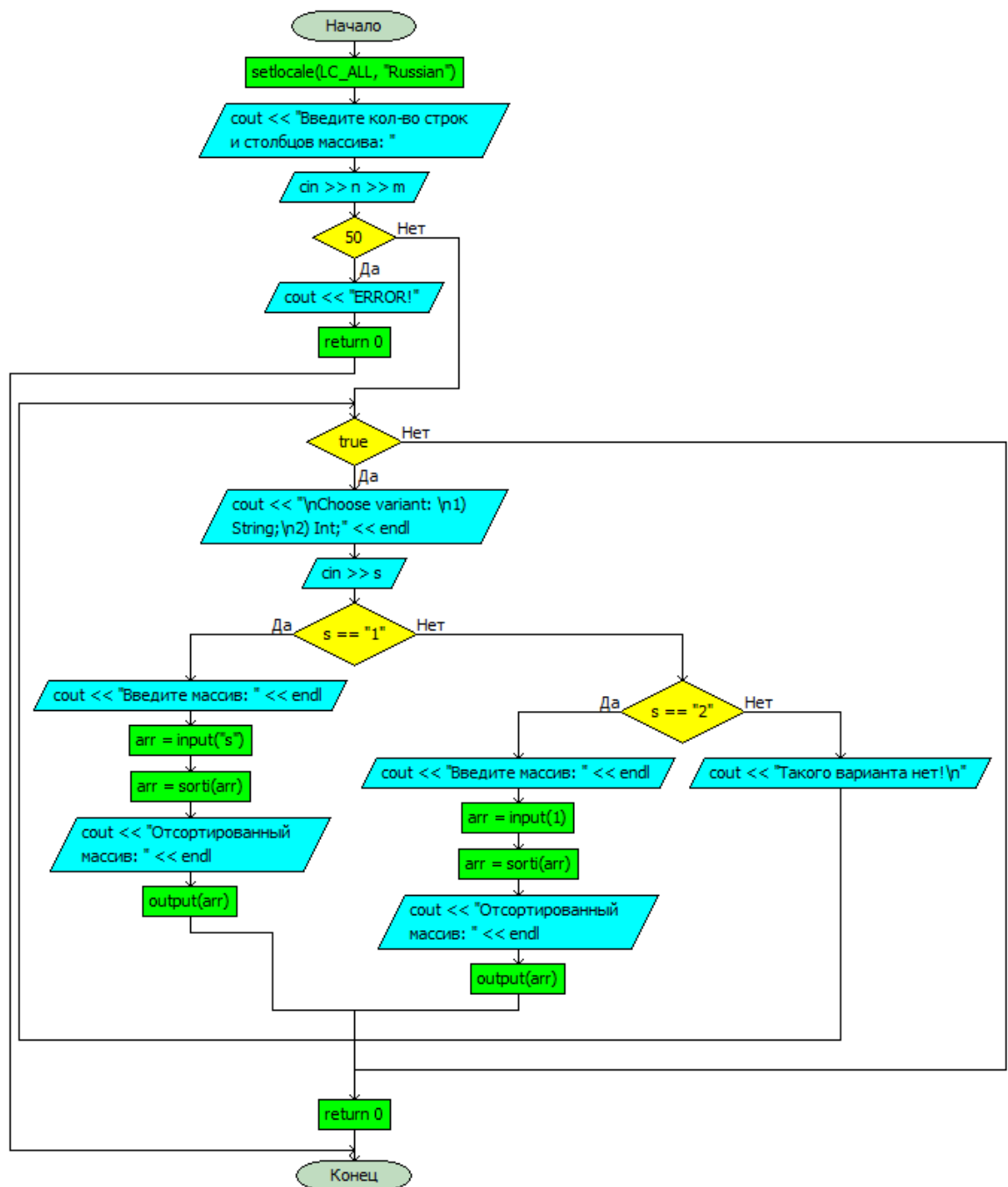


Рисунок 1.1 – Блок-схема алгоритма – функция main

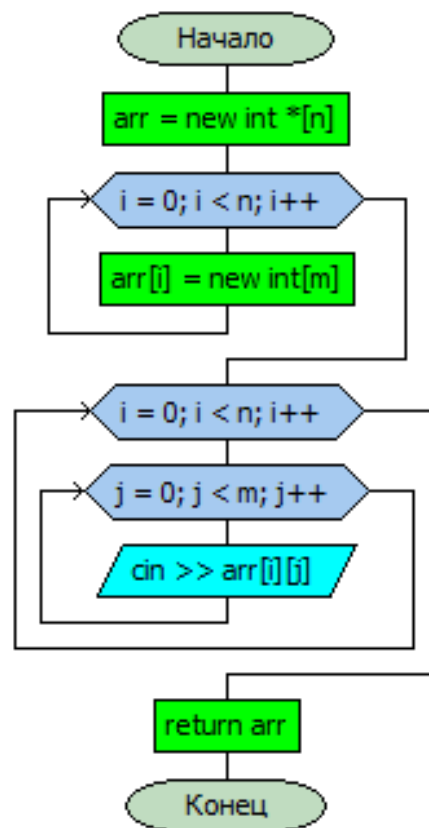


Рисунок 1.2 – Блок-схема алгоритма – функция `input` для целочисленной матрицы

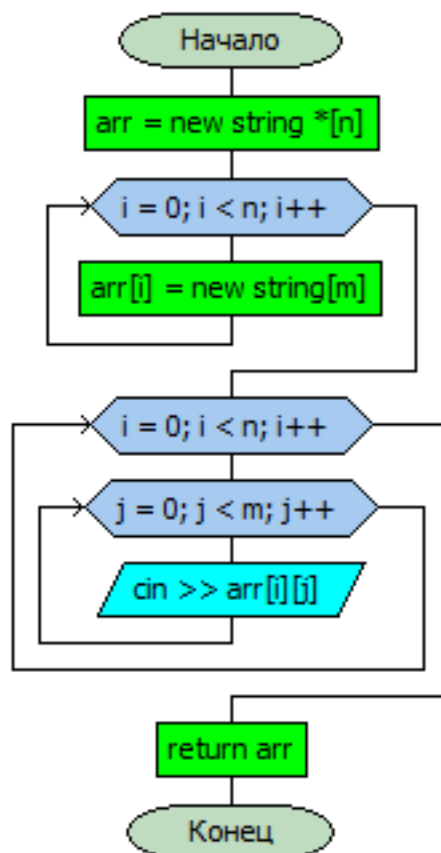


Рисунок 1.3 – Блок-схема алгоритма – функция `input` для строковой матрицы

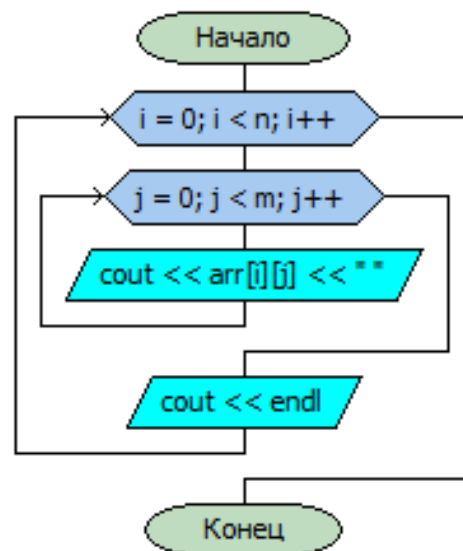


Рисунок 1.4 – Блок-схема алгоритма – функция output для целочисленной матрицы

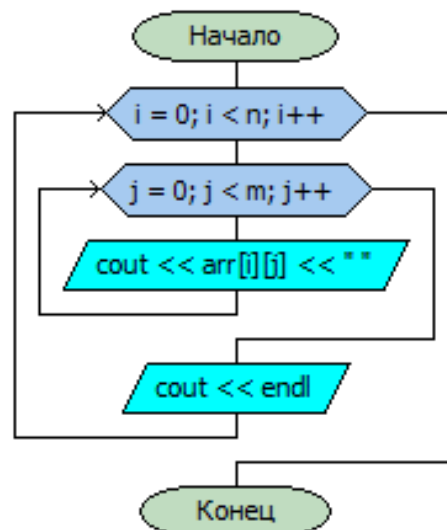


Рисунок 1.5 – Блок-схема алгоритма – функция output для строковой матрицы



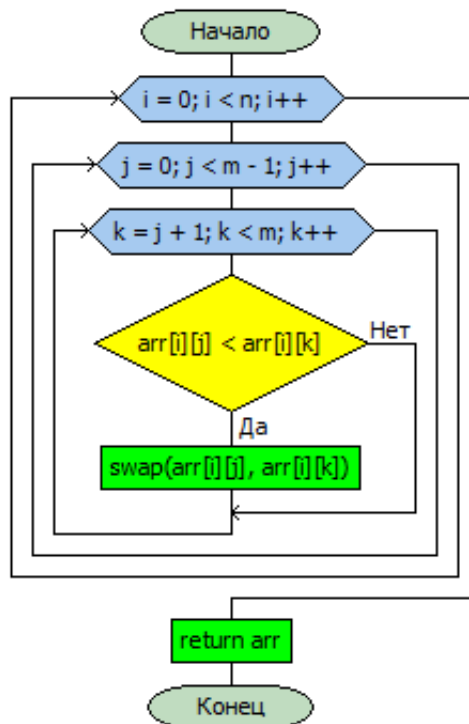


Рисунок 1.6 – Блок-схема алгоритма – функция `sorti` для целочисленной матрицы

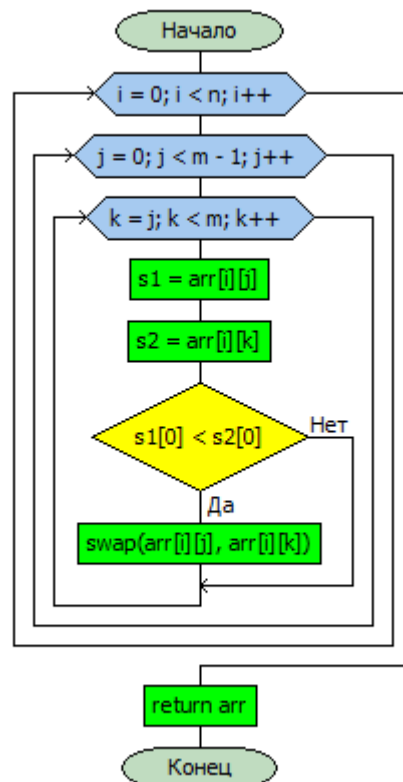


Рисунок 1.7 – Блок-схема алгоритма – функция `sorti` для строковой матрицы

Исходный код программы представлен в Листингах А.1, А.2, А.3, А.4, А.5, А.6, А.7.

### *Листинг А.1 – Процедура запуска программы*

```
#include <iostream>
#include <string>
#include <algorithm>
#include <Windows.h>

using namespace std;
int n, m;

int main()
{
    //SetConsoleCP(1251);
    //SetConsoleOutputCP(1251);
    setlocale(LC_ALL, "Russian");
    string s;
    cout << "Введите кол-во строк и столбцов массива: ";
    cin >> n >> m;
    if (n <= 0 || n > 100 || m <= 0 || m > 50) {
        cout << "ERROR!";
        return 0;
    }
    while (true) {
        cout << "\nChoose variant: \n1) String;\n2) Int;" << endl;
        cin >> s;
        if (s == "1") {
            cout << "Введите массив: " << endl;
            string** arr;
            arr = input("s");
            arr = sorti(arr);
            cout << "Отсортированный массив: " << endl;
            output(arr);
            break;
        }
        else if (s == "2") {
            cout << "Введите массив: " << endl;
            int** arr;
            arr = input(1);
            arr = sorti(arr);
            cout << "Отсортированный массив: " << endl;
            output(arr);
            break;
        }
        else {
            cout << "Такого варианта нет!\n";
        }
    }

    return 0;
}
```

### *Листинг А.2 – Процедура ввода целочисленной матрицы*

```
int** input(int a) {
    int** arr;
    arr = new int* [n];
    for (int i = 0; i < n; i++) {
        arr[i] = new int[m];
    }
    for (int i = 0; i < n; i++) {
        for (int j = 0; j < m; j++) {
            cin >> arr[i][j];
        }
    }
    return arr; }
```

*Листинг A.3 – Процедура ввода строковой матрицы*

```
string** input(string s) {
    string** arr;
    arr = new string * [n];
    for (int i = 0; i < n; i++) {
        arr[i] = new string[m];
    }
    for (int i = 0; i < n; i++) {
        for (int j = 0; j < m; j++) {
            cin >> arr[i][j];
        }
    }
    return arr;
}
```

*Листинг A.4 – Процедура вывода целочисленной матрицы*

```
void output(int** arr) {
    for (int i = 0; i < n; i++) {
        for (int j = 0; j < m; j++) {
            cout << arr[i][j] << " ";
        }
        cout << endl;
    }
}
```

*Листинг A.5 – Процедура вывода строковой матрицы*

```
void output(string** arr) {
    for (int i = 0; i < n; i++) {
        for (int j = 0; j < m; j++) {
            cout << arr[i][j] << " ";
        }
        cout << endl;
    }
}
```

*Листинг A.6 – Процедура сортировки целочисленной матрицы*

```
int** sorti(int** arr) {
    for (int i = 0; i < n; i++) {
        for (int j = 0; j < m - 1; j++) {
            for (int k = j + 1; k < m; k++) {
                if (arr[i][j] < arr[i][k]) {
                    swap(arr[i][j], arr[i][k]);
                }
            }
        }
    }
    return arr;
}
```

*Листинг A.7 – Процедура сортировки строковой матрицы*

```
string** sorti(string** arr) {
    for (int i = 0; i < n; i++) {
        for (int j = 0; j < m - 1; j++) {
            for (int k = j; k < m; k++) {
                string s1 = arr[i][j];
                string s2 = arr[i][k];
                if (s1[0] < s2[0]) {
                    swap(arr[i][j], arr[i][k]);
                }
            }
        }
    }
    return arr;
}
```

Пример работы программы представлен на рисунках [2.1](#) и [2.2](#).

```
Введите кол-во строк и столбцов массива: 3 4

Choose variant:
1) String;
2) Int;
2
Введите массив:
3 6 1 9
6 0 3 7
7 9 2 5
Отсортированный массив:
9 6 3 1
7 6 3 0
9 7 5 2
```

Рисунок 2.1 – Пример работы программы с целочисленным массивом

```
Введите кол-во строк и столбцов массива: 4 5

Choose variant:
1) String;
2) Int;
1
Введите массив:
s g c e a
f b l t w
y c a d e
y z d a r
Отсортированный массив:
s g e c a
w t l f b
y e d c a
z y r d a
```

Рисунок 2.2 – Пример работы программы с строковым массивом

## **ЗАКЛЮЧЕНИЕ**

В результате выполнения данной практической работы были закреплены основные знания о работе с целочисленными и строковыми массивами, а также их индексами, при помощи перегруженных функций. Были закреплены навыки использования основных библиотек языка программирования C++.

## СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Кубенский А.А. Структуры и алгоритмы обработки данных: объектноориентированный подход и реализация на С++ / А.А. Кубенский.— М.: БХВ-Петербург, 2017.— 300 с.
2. Стивен Прата. Язык программирования С++ (С++11). Лекции и упражнения, 6-е издание — М.: Вильямс, 2012. — 1248 с.
3. Седжвик, Р. Алгоритмы на С++ / Р. Седжвик.— М.:Вильямс, 2017.— 1056 с.