|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  |  | | --- | --- | --- | |  |  |  | | 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-01-22  *(учебная группа)* | Зырянов М.А. |
| Принял старший преподаватель | Асадова Ю.С. |

|  |  |  |
| --- | --- | --- |
| Практическая работа выполнена | «4» ноября 2023г. | *\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_*  *(подпись студента)* |
|  |  |  |
| «Зачтено» | «\_\_»\_\_\_\_\_\_\_2023г. | *\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_*  *(подпись руководителя)* |

Москва 2023

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

Институт информационных технологий

Кафедра вычислительной техники

Выполнено \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/М.А. Зырянов/

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

**Задание на практическую работу №8**

Дисциплина: «Алгоритмические основы обработки данных»

Студент Зырянов Максим Алексеевич Шифр 22И1453 Группа ИВБО-01-22

**1. Тема**: «ПЕРЕГРУЗКА ФУНКЦИЙ».

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

**3. Исходные данные:** матрица из n строк и m столбцов (n<=100, m<=50)

**4. Задание:** Упорядочить все элементы матрицы таким образом, чтобы при чтении матрицы по столбцам ее элементы образовывали отсортированный по возрастанию массив

**5. Содержание отчета:**

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

Руководитель работы Ю.С. Асадова \_\_\_\_\_\_\_\_\_\_\_\_ «\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_ 2023г.

подпись

Задание принял к исполнению М.А. Зырянов \_\_\_\_\_\_\_\_\_\_\_\_ «\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_ 2023г.

подпись

**СОДЕРЖАНИЕ**

[ВВЕДЕНИЕ 4](#_Toc145183808)

[1 ОСНОВНОЙ РАЗДЕЛ 5](#_Toc145183809)

[2 БЛОК-СХЕМА АЛГОРИТМА 7](#_Toc145183810)

[3 ИСХОДНЫЙ КОД 8](#_Toc145183811)

[4 ПРИМЕР РАБОТЫ ПРОГРАММЫ 10](#_Toc145183812)

[ЗАКЛЮЧЕНИЕ 11](#_Toc145183813)

[СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ 12](#_Toc145183814)

# ВВЕДЕНИЕ

Требуется применить на практике перегрузку функции.

Постановка задачи:

Разработать программу, которая вводит матрицу из n строк и m столбцов (n<=100, m<=50) и упорядочивает элементы матрицы. 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. Упорядочить все элементы матрицы таким образом, чтобы при чтении матрицы по столбцам ее элементы образовывали отсортированный по возрастанию массив.

# 1 ОСНОВНОЙ РАЗДЕЛ

Пользователю требуется ввести количество столбцов и строк, далее будет предоставлен выбор каким типом заполнить матрицу (1-Int, 2-Str). Далее запускаются функции ввода, сортировки и вывода матрицы. Сортировка матрицы происходит с помощью дополнительной матрицы.

# 2 БЛОК-СХЕМА АЛГОРИТМА

Представим описание алгоритма в графическом виде на рисунке 2.1.

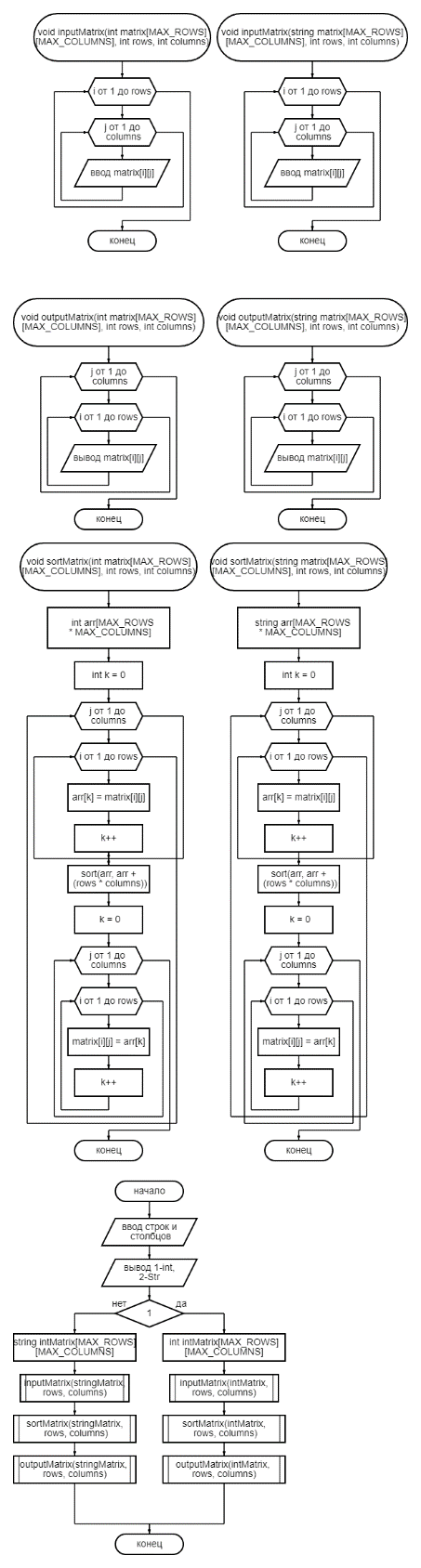


Рисунок 2.1 – Блок – схема алгоритма программы

# 3 ИСХОДНЫЙ КОД

Программная реализация алгоритма для решения задачи представлена ниже.

Листинг 3.1 – Процедура проверки попадания точки в закрашенную область

|  |
| --- |
| #include <iostream>  #include <algorithm>  #include <string>  const int MAX\_ROWS = 100;  const int MAX\_COLUMNS = 50;  void inputMatrix(int matrix[MAX\_ROWS][MAX\_COLUMNS], int rows, int columns) {  std::cout << "Enter ur matrix:\n";  for (int i = 0; i < rows; i++) {  for (int j = 0; j < columns; j++) {  std::cin >> matrix[i][j];  }  }  }  void inputMatrix(std::string matrix[MAX\_ROWS][MAX\_COLUMNS], int rows, int columns) {  std::cout << "Enter ur matrix:\n";  for (int i = 0; i < rows; i++) {  for (int j = 0; j < columns; j++) {  std::cin >> matrix[i][j];  }  }  }  void outputMatrix(int matrix[MAX\_ROWS][MAX\_COLUMNS], int rows, int columns) {  std::cout << "Matrix:\n";  for (int j = 0; j < columns; j++) {  for (int i = 0; i < rows; i++) {  std::cout << matrix[i][j] << " ";  }  std::cout << std::endl;  }  }  void outputMatrix(std::string matrix[MAX\_ROWS][MAX\_COLUMNS], int rows, int columns) {  std::cout << "Matrix:\n";  for (int j = 0; j < columns; j++) {  for (int i = 0; i < rows; i++) {  std::cout << matrix[i][j] << " ";  }  std::cout << std::endl;  }  }  void sortMatrix(int matrix[MAX\_ROWS][MAX\_COLUMNS], int rows, int columns) {  int arr[MAX\_ROWS \* MAX\_COLUMNS];  int k = 0;  for (int j = 0; j < columns; j++) {  for (int i = 0; i < rows; i++) {  arr[k] = matrix[i][j];  k++;  }  }  std::sort(arr, arr + (rows \* columns));  k = 0;  for (int j = 0; j < columns; j++) {  for (int i = 0; i < rows; i++) {  matrix[i][j] = arr[k];  k++;  }  }  }  void sortMatrix(std::string matrix[MAX\_ROWS][MAX\_COLUMNS], int rows, int columns) {  std::string arr[MAX\_ROWS \* MAX\_COLUMNS];  int k = 0;  for (int j = 0; j < columns; j++) {  for (int i = 0; i < rows; i++) {  arr[k] = matrix[i][j];  k++;  }  }  std::sort(arr, arr + (rows \* columns));  k = 0;  for (int j = 0; j < columns; j++) {  for (int i = 0; i < rows; i++) {  matrix[i][j] = arr[k];  k++;  }  }  }  int main() {  int rows, columns;  std::cout << "Enter size of matrix";  std::cin >> rows >> columns;  int choice;  std::cout << "Enter type (1 - Int, 2 - Str): ";  std::cin >> choice;  if (choice == 1) {  int intMatrix[MAX\_ROWS][MAX\_COLUMNS];  inputMatrix(intMatrix, rows, columns);  sortMatrix(intMatrix, rows, columns);  outputMatrix(intMatrix, rows, columns);  }  else if (choice == 2) {  std::string stringMatrix[MAX\_ROWS][MAX\_COLUMNS];  inputMatrix(stringMatrix, rows, columns);  sortMatrix(stringMatrix, rows, columns);  outputMatrix(stringMatrix, rows, columns);  }  } |

# 4 ПРИМЕР РАБОТЫ ПРОГРАММЫ

Пример работы программы, когда входные данные 4,5,6,3,2,1,9,7,8 представлен на рисунке 4.1.

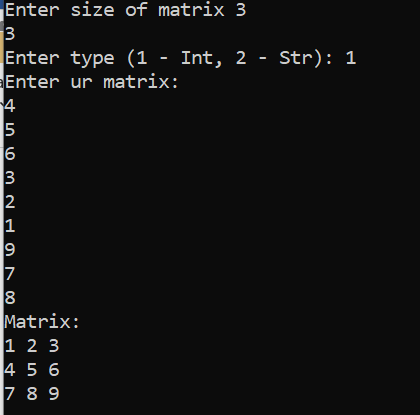


Рисунок 4.1 – Пример работы программы – целочисленная матрица

Пример работы программы, когда входные данные f,d,q,p,d,x,c,z,a представлен на рисунке 4.1.

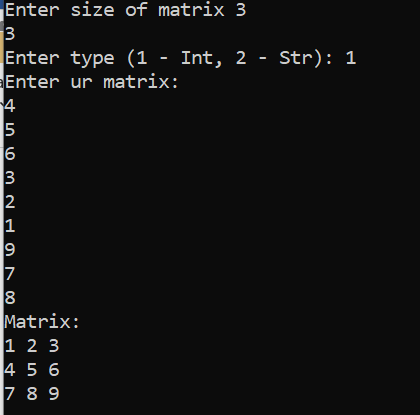


Рисунок 4.1 – Пример работы программы – строковая матрица

# ЗАКЛЮЧЕНИЕ

В ходе выполнения данной практической работы была реализована программа с перегруженными функциями. Также были приобретены навыки работы с сортировкой на языке программирования C++.

# СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Лозовский В.В. Алгоритмические основы обработки данных: учебное пособие / Лозовский В.В., Платонова О.В., Штрекер Е.Н. — М.: МИРЭА – Российский технологический университет, 2022. – 337 с.

2. Платонова О.В. Алгоритмические основы обработки данных: методические указания / Платонова О.В., Асадова Ю.С., Расулов М.М. — М.: МИРЭА – Российский технологический университет, 2022. — 73 с.

3. Белик А.Г. Алгоритмы и структуры данных: учебное пособие / А.Г. Белик, В.Н. Цыганенко. — Омск: ОмГТУ, 2022. — 104 с. — ISBN 978-5-8149-3498-7. — Текст: электронный // Лань: электронно-библиотечная система. — URL: https://e.lanbook.com/book/343688 (дата обращения: 04.11.2023)

4. Павлов Л.А. Структуры и алгоритмы обработки данных / Л.А. Павлов, Н.В. Первова. — 2-е изд., стер. — Санкт-Петербург: Лань, 2022. — 256 с. — ISBN 978-5-507-44105-1. — Текст: электронный // Лань: электронно-библиотечная система. — URL: https://e.lanbook.com/book/207563 (дата обращения: 04.11.2023)

5. Пантелеев Е.Р. Алгоритмы и структуры данных: учебное пособие / Е.Р. Пантелеев, А.Л. Алыкова. — Иваново: ИГЭУ, 2018. — 142 с. — Текст: электронный // Лань: электронно-библиотечная система. — URL: https://e.lanbook.com/book/154576 (дата обращения: 04.11.2023)