



**«Московский государственный технический университет
имени Н.Э. Баумана»
(национальный исследовательский университет)
(МГТУ им. Н.Э. Баумана)**

ФАКУЛЬТЕТ ИНФОРМАТИКА И СИСТЕМЫ УПРАВЛЕНИЯ
КАФЕДРА КОМПЬЮТЕРНЫЕ СИСТЕМЫ И СЕТИ (ИУ6)

О т ч е т

по лабораторной работе № 5

Название лабораторной работы: Матрицы.

Дисциплина: Алгоритмизация и программирование.

Студент гр. ИУ6-14Б

(Подпись, дата)

21.09.2024 Д.А.Пасхальная

(И.О. Фамилия)

Преподаватель

(Подпись, дата)

21.09.2024 О.А.Веселовская

(И.О. Фамилия)

Цель работы – Решить поставленную задачу, используя средства управления вводом/выводом. Решить, используя подпрограмму.

Задание – Дан трехмерный массив $R(10,10,2)$. Каждым элементом первого квадрата 10×10 является символ «!», каждым элементом второго – символ «?». Вывести массив на экран в виде прямоугольника так, чтобы столбец восклицательных знаков чередовался со столбцом вопросительных. Из вещественной матрицы $E(n,m)$, $n \leq 8$, $m \leq 12$, удалить строку, содержащую наибольший элемент. Вывести на экран исходную и полученную матрицы, а также найденный элемент и его координаты.

Ход работы:

- Изображение схемы алгоритмов программы.
- Написание программы.
- Тестирование программы при различных тестовых данных.
- Вывод.

Для начала составим схему алгоритма программы:

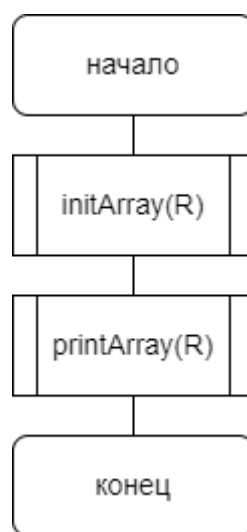
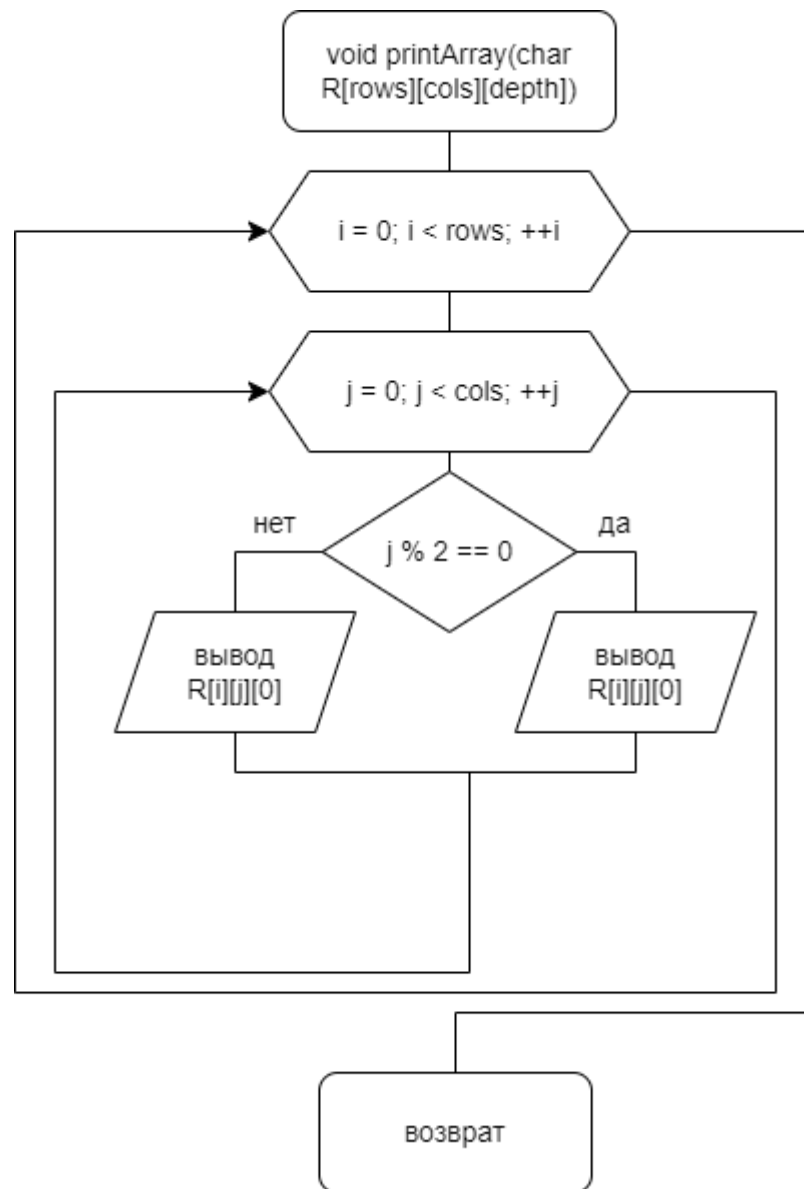


Рисунок 1 – Схема алгоритма программы 1



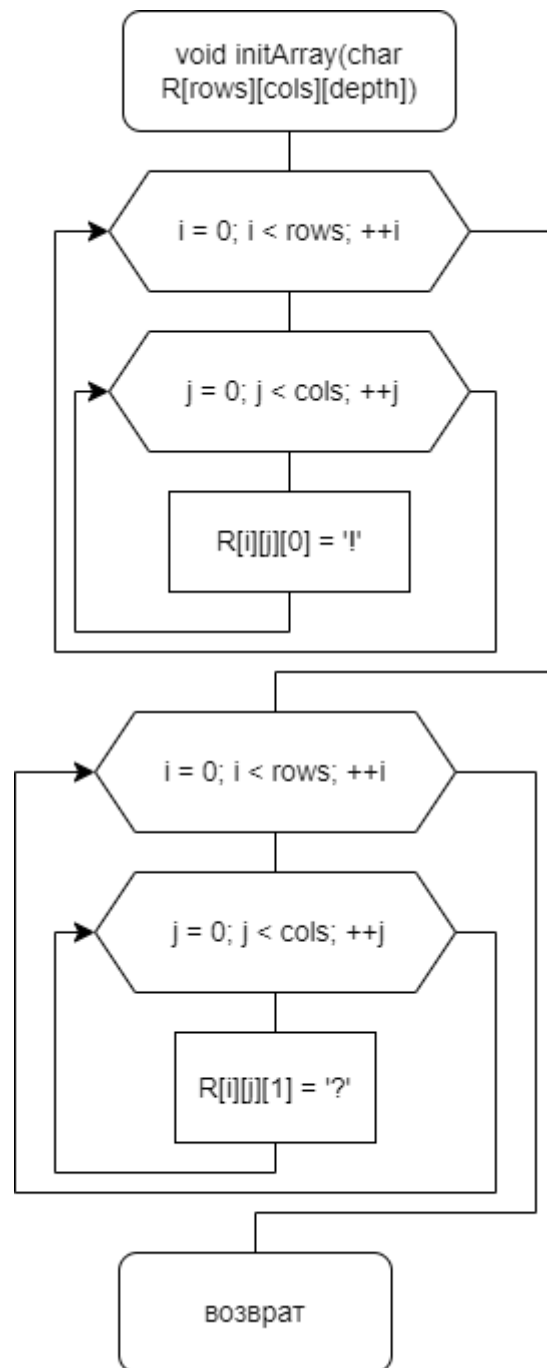


Рисунок 2 – Схемы алгоритмов подпрограмм, используемых в программе 1

Напишем код программы.

```

1  #include <iostream>
2  using namespace std;
3
4  const int rows = 10;
5  const int cols = 10;
6  const int depth = 2;
7
8  void initArray(char R[rows][cols][depth]) {
9      for (int i = 0; i < rows; ++i) {
10         for (int j = 0; j < cols; ++j) {
11             R[i][j][0] = '!';
12         }
13     }
14     for (int i = 0; i < rows; ++i) {
15         for (int j = 0; j < cols; ++j) {
16             R[i][j][1] = '?';
17         }
18     }
19 }
20
21 void printArray(char R[rows][cols][depth]) {
22     for (int i = 0; i < rows; ++i) {
23         for (int j = 0; j < cols; ++j) {
24             if (j % 2 == 0) {
25                 cout << R[i][j][0];
26             }
27             else {
28                 cout << R[i][j][1];
29             }
30         }
31     }
32     cout << endl;
33 }
34
35
36 int main() {
37     char R[rows][cols][depth];
38     initArray(R);
39     printArray(R);
40
41     return 0;
42 }

```

Рисунок 3 – Программа для решения первой части задания

[illegible]

Рисунок 4 – Вывод программы

Программа работает верно на введенных тестовых данных.

Далее составим схему алгоритмов и напомним код программы для второй части задания:

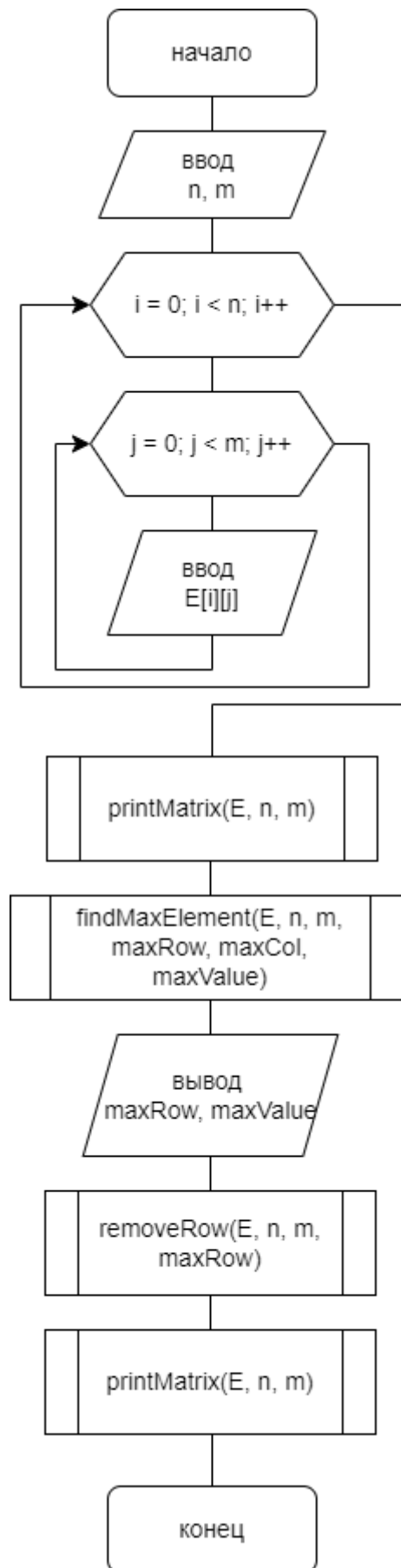
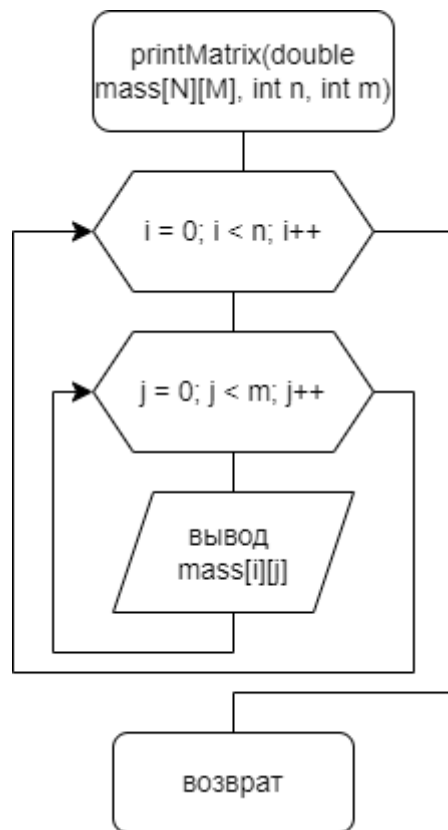
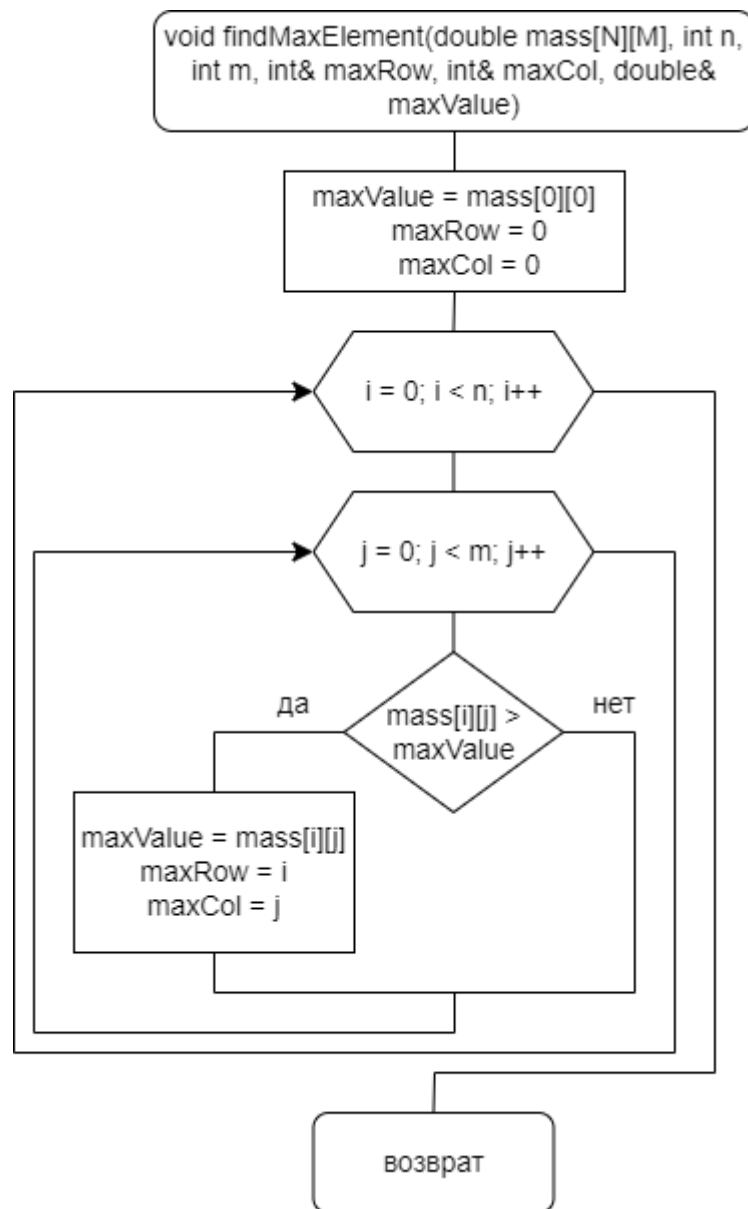


Рисунок 5 – Схема алгоритма программы 2





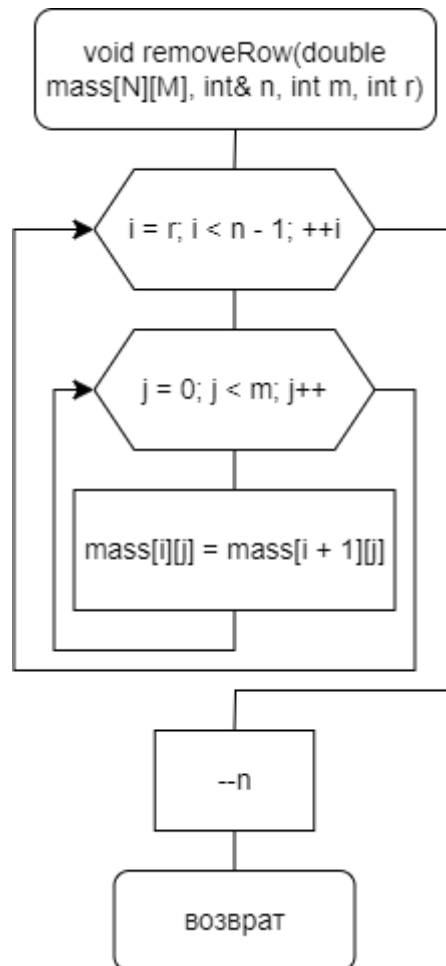


Рисунок 6 – Схемы алгоритмов подпрограмм, используемых в программе 2

```

1  #include <iostream>
2  #include <iomanip>
3
4  using namespace std;
5  const int N = 8;
6  const int M = 12;
7
8  void printMatrix(double mass[N][M], int n, int m) {
9      for (int i = 0; i < n; ++i) {
10         for (int j = 0; j < m; ++j) {
11             cout << setw(2) << mass[i][j];
12         }
13         cout << endl;
14     }
15 }
16
17 void findMaxElement(double mass[N][M], int n, int m, int& maxRow, int& maxCol, double& maxValue) {
18     maxValue = mass[0][0];
19     maxRow = 0;
20     maxCol = 0;
21
22     for (int i = 0; i < n; ++i) {
23         for (int j = 0; j < m; ++j) {
24             if (mass[i][j] > maxValue) {
25                 maxValue = mass[i][j];
26                 maxRow = i;
27                 maxCol = j;
28             }
29         }
30     }

```

```

31 }
32
33 void removeRow(double mass[N][M], int& n, int m, int r) {
34     for (int i = r; i < n - 1; ++i) {
35         for (int j = 0; j < m; ++j) {
36             mass[i][j] = mass[i + 1][j];
37         }
38     }
39     --n;
40 }
41
42 int main() {
43     setlocale(LC_ALL, "Russian");
44     int n, m;
45     double E[N][M];
46
47     cout << "Введите размерность матрицы (n, m): ";
48     cin >> n >> m;
49
50     cout << "Введите элементы матрицы:\n";
51     for (int i = 0; i < n; i++) {
52         for (int j = 0; j < m; j++) {
53             cin >> E[i][j];
54         }
55     }
56
57     cout << "\nИсходная матрица:\n";
58     printMatrix(E, n, m);
59
60     int maxRow, maxCol;

```

```

61     double maxValue;
62     findMaxElement(E, n, m, maxRow, maxCol, maxValue);
63
64     cout << "\nМаксимальный элемент: " << maxValue << endl;
65     cout << "Координаты: (" << maxRow + 1 << ", " << maxCol + 1 << ")" << endl;
66     removeRow(E, n, m, maxRow);
67
68     cout << "\nПолученная матрица:\n";
69     printMatrix(E, n, m);
70
71     return 0;
72 }

```

Рисунок 7 – Код программы часть 2

```

Введите размерность матрицы (n, m): 2 3
Введите элементы матрицы:
4 7 8
2 3 1

Исходная матрица:
4 7 8
2 3 1

Полученная матрица:
2 3 1

Максимальный элемент: 8
Координаты: (0, 2)

```

Рисунок 8 – Тестирование программы часть 1

```
Введите размерность матрицы (n, m): 3 4
Введите элементы матрицы:
6 4 3 2
9 7 3 5
0 8 2 3

Исходная матрица:
6 4 3 2
9 7 3 5
0 8 2 3

Полученная матрица:
6 4 3 2
0 8 2 3

Максимальный элемент: 9
Координаты: (1, 0)
```

Рисунок 9 – Тестирование программы часть 2

Можем убедиться, что программа работает верно на введенных тестовых данных, находит и удаляет строку с максимальным числом вхождений.

Вывод: В ходе лабораторной работы я научилась работать с матрицами и их итерированием, использовать подпрограммы для выполнения поставленной задачи, чтобы упростить чтение и отладку программы.