



**«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)


**Отчет**

**по лабораторной работе № 5**

**Название лабораторной работы: Матрицы.**

**Дисциплина: Алгоритмизация и программирование**

Студент гр. ИУ6-14Б

 **28.10.2023**

(Подпись, дата)

**А.С. Воеводин**

(И.О. Фамилия)

Преподаватель

 **28.10.2023**

(Подпись, дата)

**О.А. Веселовская**

(И.О. Фамилия)

Москва, 2023

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

**Задание** – Дан трехмерный массив символов  $L(8,6,4)$ . Вывести его элементы на экран строками по 12 символов в виде параллелограмма. Дана целочисленная матрица  $R(n,n)$ ,  $n < 11$  и целые числа  $p$  и  $q$ . Определить сколько элементов, равных элементу  $grq$ , содержит матрица. Вывести на печать исходную матрицу, найденные элементы и их номера строки и столбца. Удалить строку с элементом  $grq$ . Вывести на экран полученную матрицу.

**Ход работы:**

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

Для начала напишем код программы:

```

const int n = 8, m = 6, k = 4;
char arr[n][m][k] = {};

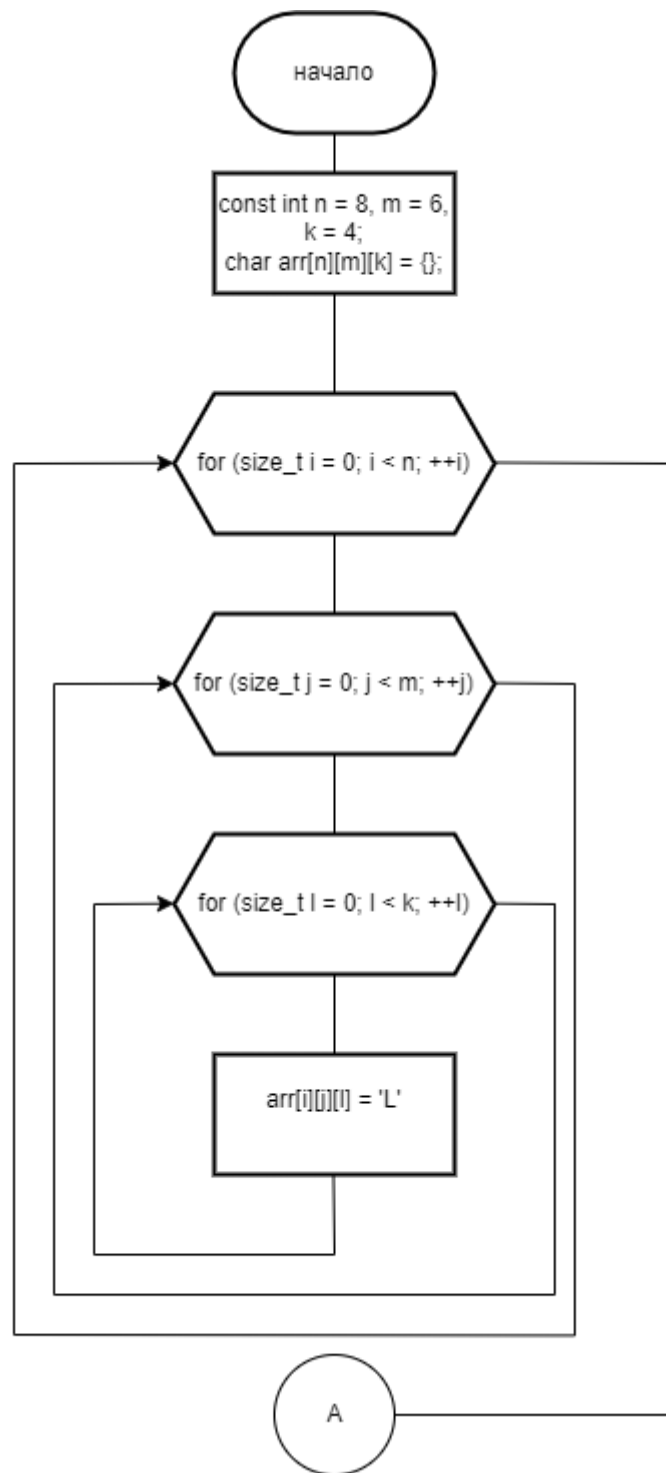
for (size_t i = 0; i < n; ++i) {
    for (size_t j = 0; j < m; ++j) {
        for (size_t l = 0; l < k; ++l) {
            arr[i][j][l] = 'L';
        }
    }
}

size_t count = 0;
size_t spaces = 1;

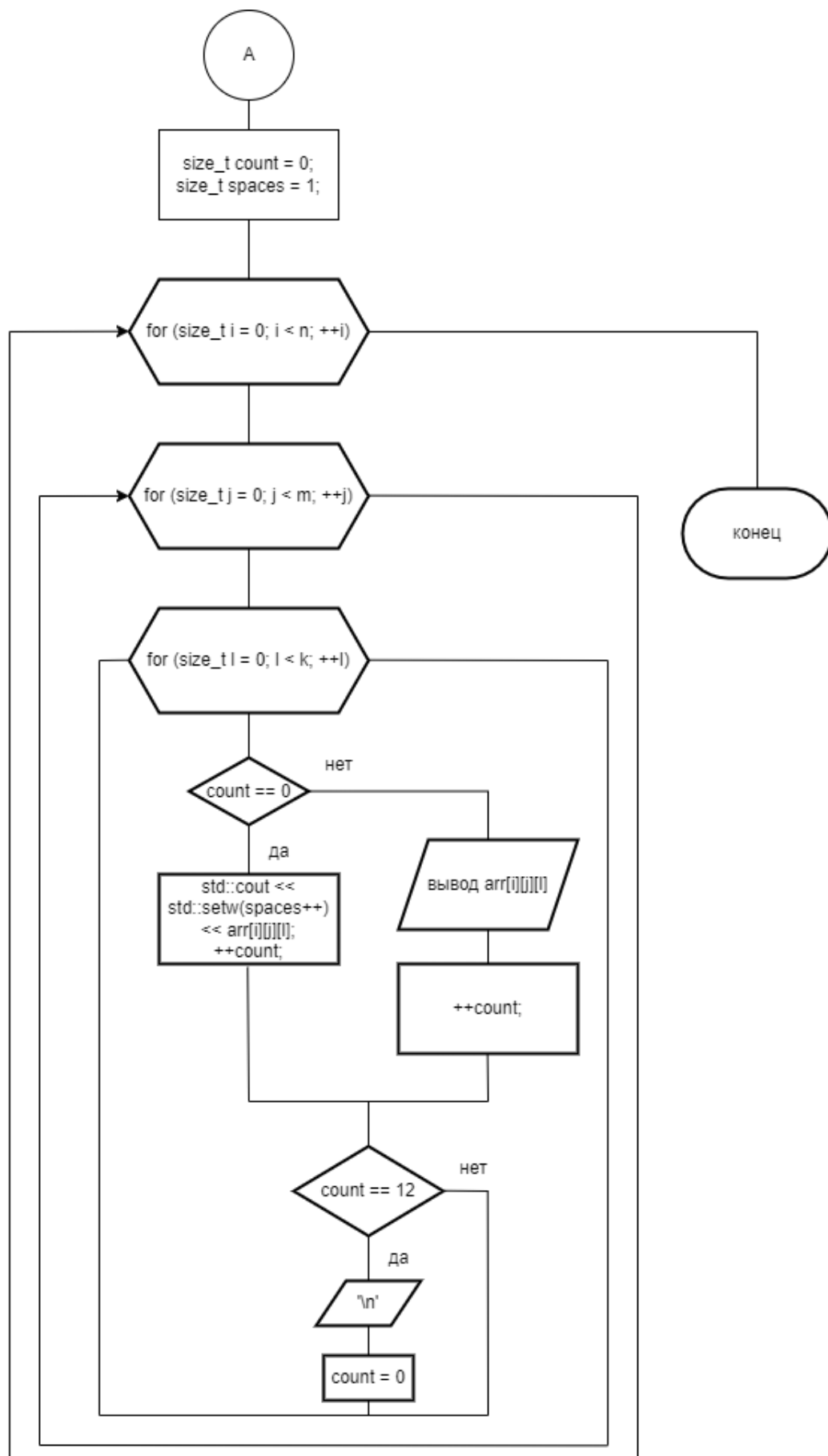
for (size_t i = 0; i < n; ++i) {
    for (size_t j = 0; j < m; ++j) {
        for (size_t l = 0; l < k; ++l) {
            if (count == 0) {
                std::cout << std::setw(spaces++) << arr[i][j][l];
                ++count;
            }
            else {
                std::cout << arr[i][j][l];
                ++count;
            }
            if (count == 12) {
                std::cout << std::endl;
                count = 0;
            }
        }
    }
}

```

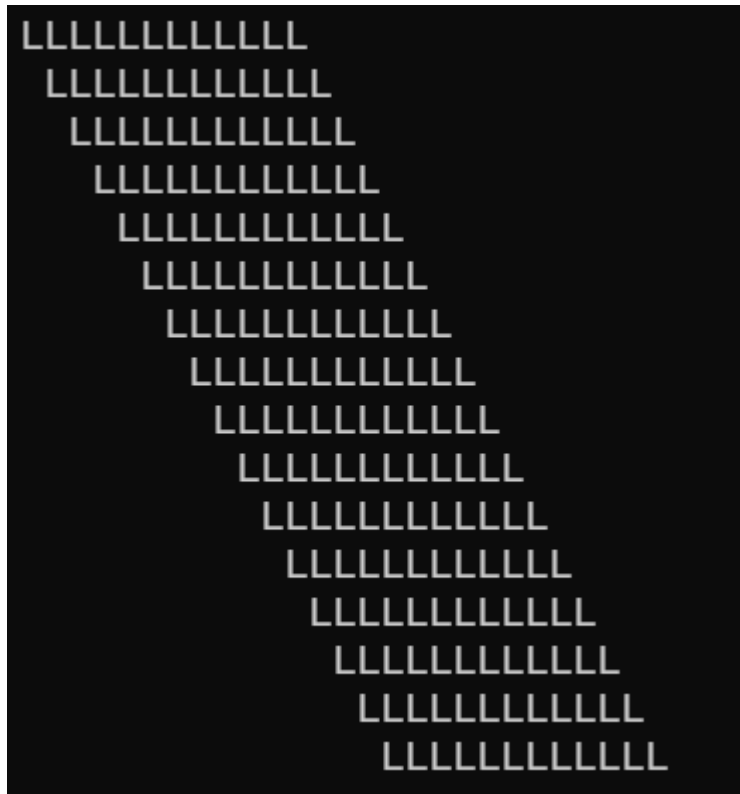
Рисунок 1 – Программа для решения первой части задания



**Рисунок 2** – Заполнение трёхмерного массива



**Рисунок 3** – Схема алгоритма программы



**Рисунок 4** – Вывод программы

Несложно посмотреть, что программа работает верно.

Далее напишем код программы для второй части задания:

```

#include <iostream>
#include <iomanip>

void Print_array(int** ptr, int n) {
    for (size_t i = 0; i < n; ++i) {
        if (ptr[i] != nullptr) {
            for (size_t j = 0; j < n; ++j) {
                std::cout << ptr[i][j] << ' ';
            }
            std::cout << std::endl;
        }
    }
}

void Fill_array(int** ptr, int n) {
    for (size_t i = 0; i < n; ++i) {
        if (ptr[i] != nullptr) {
            for (size_t j = 0; j < n; ++j) {
                std::cin >> ptr[i][j];
            }
        }
    }
}

```

Рисунок 5 – Процедуры, используемые в программе

```

int n = 0;
std::cout << "Enter n: ";
std::cin >> n;
int** arr = new int*[n];

for (size_t i = 0; i < n; ++i) {
    arr[i] = new int[n];
}

std::cout << "Enter your matrix: ";

Fill_array(arr, n);

int p = 0, q = 0;
std::cout << "Enter p and q: ";
std::cin >> p >> q;
size_t count = 0;

std::cout << "Find " << arr[p][q] << std::endl;

for (size_t i = 0; i < n; ++i) {
    for (size_t j = 0; j < n; ++j) {
        if (arr[i][j] == arr[p][q]) {
            ++count;
            std::cout << i << ' ' << j << std::endl;
        }
    }
}

```

Рисунок 6 – Код программы часть 1



```

std::cout << "Count of this value in matrix : ";
std::cout << count << std::endl;

delete[] arr[p];
arr[p] = nullptr;

std::cout << "Matrix without string with this element:\n";
Print_array(arr, n);

for (size_t i = 0; i < n; ++i) {
    if (arr[i] != nullptr) {
        delete[] arr[i];
        arr[i] = nullptr;
    }
}

delete[] arr;
arr = nullptr;
return 0;
}

```

Рисунок 7 – Код программы часть 2

Изобразим алгоритм в в виде схемы:

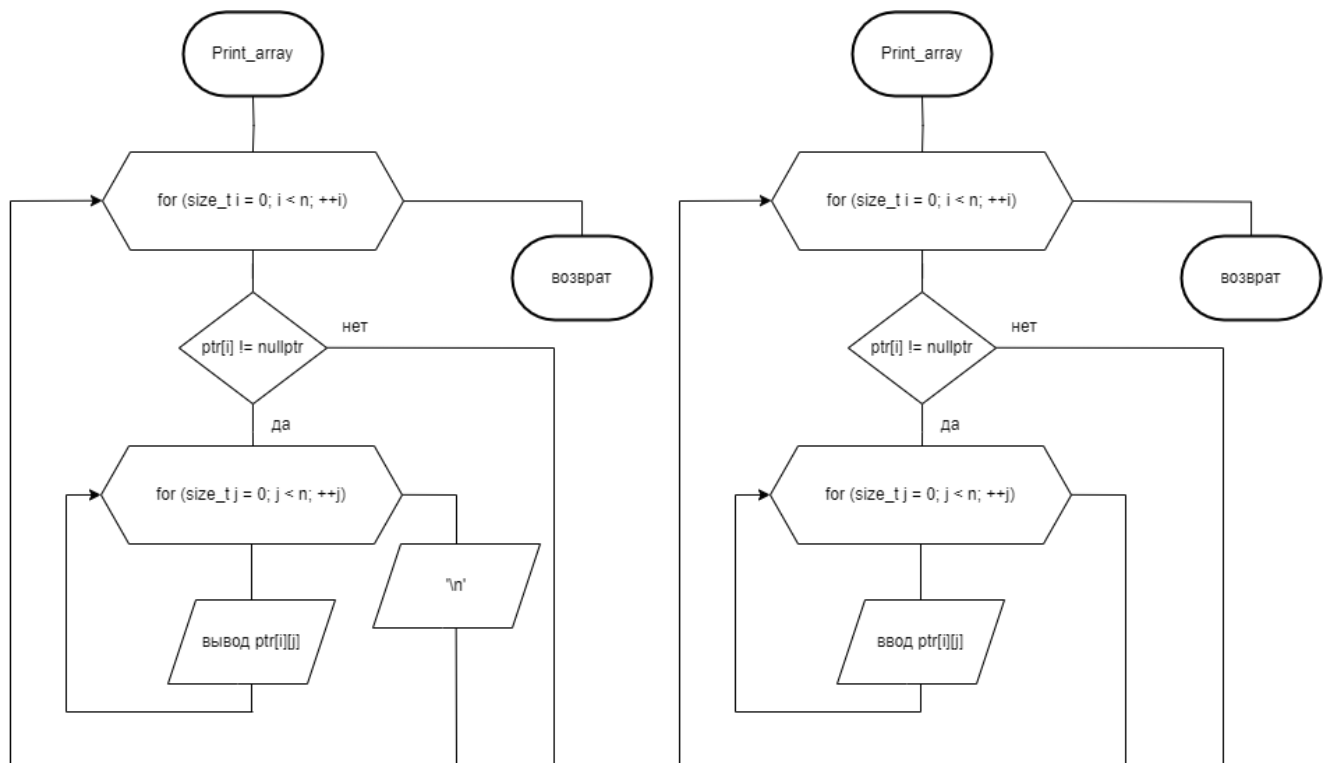
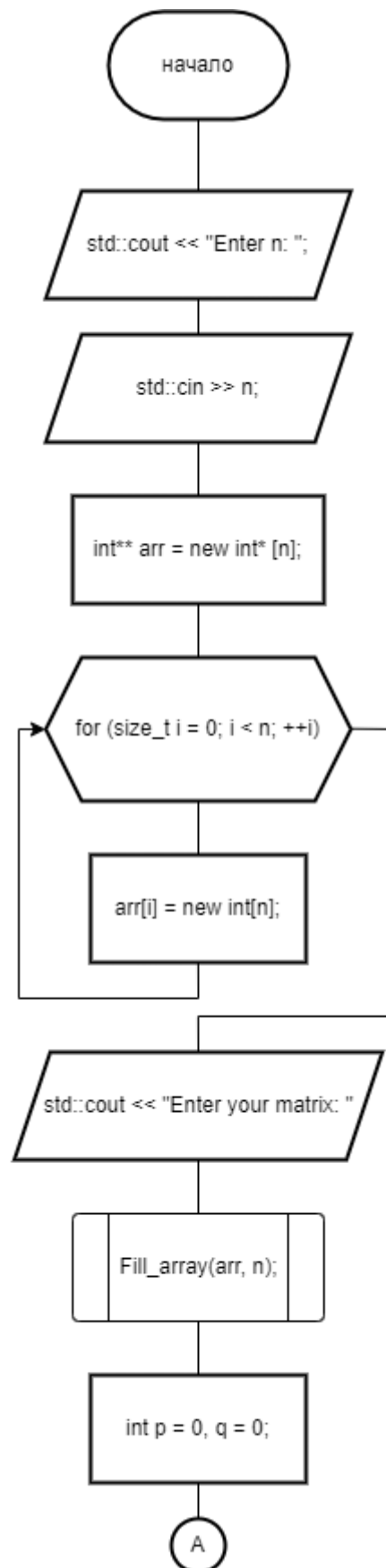


Рисунок 8 – Схемы алгоритмов процедур, используемых в программе



**Рисунок 9** – Схема алгоритма основной программы часть 1

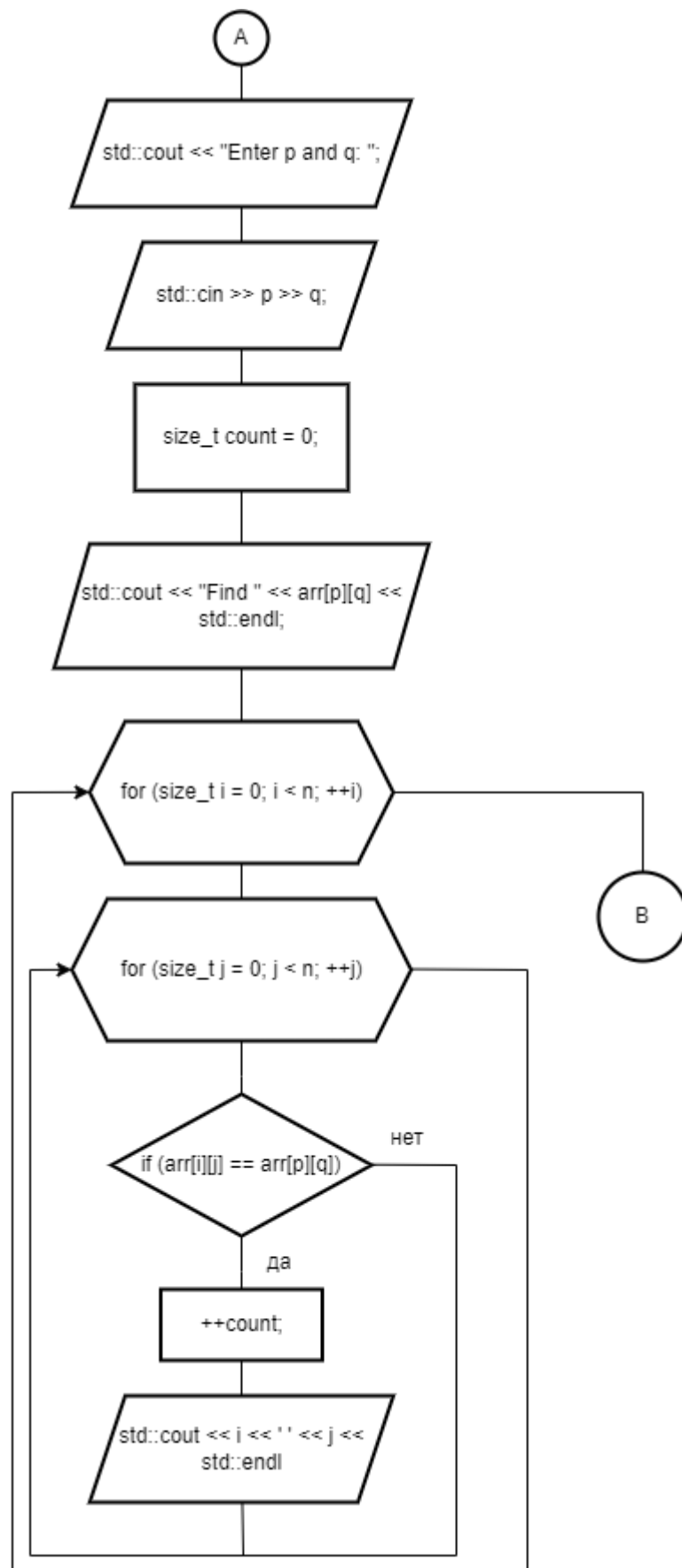
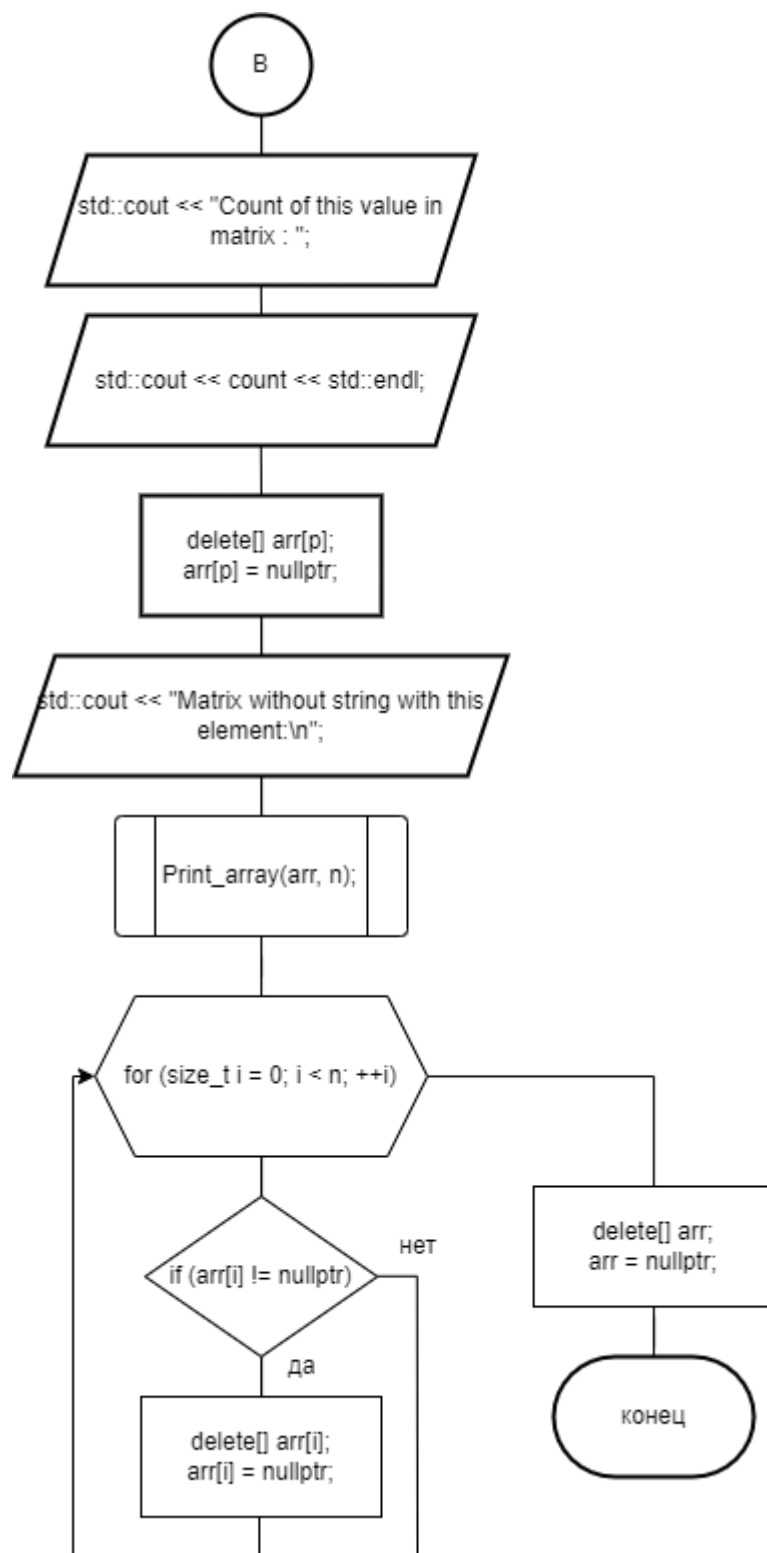


Рисунок 10 – Схема алгоритма основной программы часть 2



**Рисунок 11** – Схема алгоритма основной программы часть 3

```
Enter n: 4
Enter your matrix:
1 2 3 4
5 6 7 8
9 10 11 12
13 14 15 16
Enter p and q: 3 3
Find 16
3 3
Count of this value in matrix : 1
Matrix without string with this element:
1 2 3 4
5 6 7 8
9 10 11 12
```

Рисунок 12 – Пример использования программы часть 1

```
Enter n: 4
Enter your matrix:
1 2 3 3
4 5 6 3
5 5 5 5
1 1 1 1
Enter p and q: 0 2
Find 3
0 2
0 3
1 3
Count of this value in matrix : 3
Matrix without string with this element:
4 5 6 3
5 5 5 5
1 1 1 1
```

Рисунок 13 – Пример использования программы часть 2

Можем убедиться, что программа работает верно, находит количество вхождений заданного элемента и выводит матрицу без строки, в которой этот элемент находится.

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