

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ  
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ  
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ  
«САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ  
АЭРОКОСМИЧЕСКОГО ПРИБОРОСТРОЕНИЯ»

КАФЕДРА АЭРОКОСМИЧЕСКИХ КОМПЬЮТЕРНЫХ И ПРОГРАММНЫХ СИСТЕМ

ОЦЕНКА ДОКЛАДА:

РУКОВОДИТЕЛЬ:

Кандидат техн. наук, доцент  
должность, уч. степень, звание

\_\_\_\_\_  
подпись, дата

Л.Н.Бариков  
инициалы, фамилия

ОТЧЁТ О ЛАБОРАТОРНОЙ РАБОТЕ № 10

по курсу: ОСНОВЫ ПРОГРАММИРОВАНИЯ  
на тему: «МЕТОДЫ СОРТИРОВКИ»

РАБОТУ ВЫПОЛНИЛ

СТУДЕНТ ГР. №

4143

\_\_\_\_\_  
подпись, дата

Е.Д.Тегай  
инициалы, фамилия

## Цель работы

Изучение методов сортировки структур данных; совершенствование навыков процедурного программирования на языке C/C++ при решении задач обработки статических и динамических массивов.

## Задание на программирование

Используя технологию процедурного программирования, реализовать заданный метод сортировки и применить его для указанных фрагментов числовой матрицы в соответствии с индивидуальным заданием.

## Индивидуальное задание

Сортировка по возрастанию методом вставки. Область сортировки показана на рисунке 1.

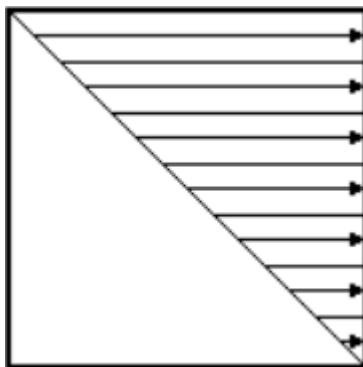


Рисунок 1 – Область сортировки

## Порядок выполнения работы

1. Получить у преподавателя индивидуальное задание: метод сортировки и вид сортируемых фрагментов матрицы. В зависимости от задания стрелка на рисунке указывает направление возрастания или убывания значений элементов фрагментов матрицы после сортировки.
2. Исходная матрица размера  $2n \times 2n$  (в вариантах 33 – 40 размер матрицы  $(2n + 1) \times (2n + 1)$ ) не должна содержать элементов с одинаковыми значениями.
3. Разработать математическую модель: описать с помощью формул и рисунков структуру массива и процесс его преобразования. У результирующей матрицы должны быть отсортированы значения элементов заданных фрагментов и обнулены все остальные.

4. Провести структуризацию задачи. С этой целью выделить подзадачи, которые будут реализованы в виде отдельных функций (ввод исходных данных, вывод содержимого массива, обработка массива в соответствии с заданием и т. д.). При этом запрещается совмещать в одной функции решение нескольких подзадач.
5. Обязательное требование – введение собственных типов данных, логически точно разделяющих и группирующих информацию, используемую при решении задачи (тип индекса элемента массива, тип значения элемента массива, тип массива и т. п.).
6. Построить схему алгоритма функции сортировки.
7. Составить программу на языке C/C++, реализовав при этом функции ввода исходной матрицы, сортировки значений элементов заданных фрагментов, обнуления значений элементов не сортируемых фрагментов и вывода значений элементов матрицы.
8. Входные данные на этапах тестирования и демонстрации работы преподавателю должны задаваться либо с использованием специально подобранных арифметических формул, либо вводиться с клавиатуры по запросу. **Датчики псевдослучайных чисел использовать запрещается.**
9. Выходные данные должны выводиться на экран с пояснениями. Вывод исходной и преобразованной матриц должен быть реализован в виде отдельной функции, вызов которой осуществляется из функции *main()*.
10. Проверить и продемонстрировать преподавателю работу программы на полном наборе тестов. Входные и выходные массивы должны выводиться в одном и том же формате.
11. Использовать стандартные потоковые объекты ввода/вывода *cin* и *cout*.
12. Оформить отчет о работе в составе: постановка задачи, схема алгоритма функции сортировки, описание процесса решения задачи, текст программы, контрольные примеры (скриншоты).
13. Текст программы в отчете не должен представлять из себя скриншот.

14. Скриншоты тестов должны легко читаться. Все их неинформативные части должны быть удалены.

### Описание решения

Исходная матрица, содержащая  $2n$  строк и  $2n$  столбцов, не должна содержать элементов с одинаковыми значениями.

Сортировка по возрастанию методом вставки.

### Текст программы

```
//Методы сортировки. Массивы динамические двумерные
//Сортировка элементов всех строк двумерного массива
//по возрастанию методом вставки

#include <iostream>
#include <locale>
#include <iomanip> // эта библиотека позволяет использовать setw

using namespace std;

typedef int telem; //определение типа значений элементо массива
typedef telem* tstr; //определение типа "указатель на telem"
typedef tstr* tmas; //определение типа "указатель на указатель на telem"

//функция, позволяющая заполнить массив
void filling_mas(tmas a, int number);
//функция, сортирующая массив
void sort_mas(tmas a, int number);
//функция, выводящая массив
void out_mas(tmas a, int number);
//функция, обнуляющая неотсортированные элементы массива
void obnulenie(tmas a, int number);

//заполнение массива
void filling_mas(tmas a, int number) {
    cout << "Заполнение массива:" << endl;
    cout << "Пожалуйста, введите значения элементов массива размером ";
    cout << 2 * number << " через пробел, а затем нажмите <Enter>" << endl;
    for (int i = 0; i < 2 * number; i++) {
        for (int j = 0; j < 2 * number; j++) {
            cin >> a[i][j];
        }
    }
}

//вывод массива
void out_mas(tmas a, int number) {
    for (int i = 0; i < 2 * number; i++) {
        for (int j = 0; j < 2 * number; j++) {
            cout << setw(number) << a[i][j] << " ";
        }
        cout << endl;
    }
}

//сортировка массива
void sort_mas(tmas a, int number) {
    int k, l;
    telem z;
    int n = 2 * number;

    for (int i = 0; i < n; i++) {
        for (int j = i; j < n; j++) {
            z = a[i][j]; //выделение текущего элемента
            l = i;
            // поиск места вставки
```

```

        while (1 < j && a[i][1] < z) {
            l++;
        }
        for (k = j - 1; k >= 1; k--)
            a[i][k + 1] = a[i][k]; // сдвиг элементов
        a[i][1] = z;
    }
}

//обнуление неотсортированных элементов
void obnulenie(tmas a, int number)
{
    for (int j = 0; j < 2 * number; j++) {
        for (int i = j + 1; i < 2 * number; i++) {
            a[i][j] = 0;
        }
    }
}

int main()
{
    setlocale(LC_ALL, "Rus");

    int number; // создание переменной number, отвечающей за размер матрицы
    tmas a; //объявление переменной-указателя на указатель на telem

    cout << "Пожалуйста, введите значение, равное размеру искомой квадратной матрицы: ";
    cin >> number; // ввод пользователем размера матрицы
    cout << endl << endl;

    a = new tstr[2 * number]; //выделение дин. памяти под массив
    for (int i = 0; i < 2 * number; i++) { // выделение памяти под каждую строку
        a[i] = new telem[2 * number]; /* каждому элементу массива указателей на строки
                                         присваивается адрес
начала области памяти,                                         выделяемой под строку */
    }

    filling_mas(a, number);
    cout << endl;
    cout << "Ваша матрица имеет вид: " << endl;
    out_mas(a, number);
    cout << endl;
    sort_mas(a, number);
    obnulenie(a, number);
    cout << "Ваша матрица после сортировки: " << endl;
    out_mas(a, number);

    cout << "Для завершения нажмите <Enter>";

    // освобождение динамической памяти
    for (int i = 0; i < number; i++) {
        delete[] a[i];
    }
    delete[] a;
    return 0;
}

```

## Результаты работы

На рисунках 7 – 8 показаны результаты работы программы.

```
Пожалуйста, введите значение, равное размеру искомой квадратной матрицы: 2

Заполнение массива:
Пожалуйста, введите значения элементов массива размером 4 через пробел, а затем нажмите <Enter>
16 15 14 13
12 11 10 9
8 7 6 5
4 3 2 1

Ваша матрица имеет вид:
16 15 14 13
12 11 10 9
8 7 6 5
4 3 2 1

Ваша матрица после сортировки:
13 14 15 16
0 9 10 11
0 0 5 6
0 0 0 1
```

Рисунок 7 – Результаты работы программы

```
пожалуйста, введите значение, равное размеру искомой квадратной матрицы: 3

Заполнение массива:
Пожалуйста, введите значения элементов массива размером 6 через пробел, а затем нажмите <Enter>
1 5 9 7 2 4
7 2 9 3 5 1
3 9 5 2 3 5
7 9 3 2 1 6
6 3 2 4 6 8
9 9 4 5 3 2

Ваша матрица имеет вид:
1 5 9 7 2 4
7 2 9 3 5 1
3 9 5 2 3 5
7 9 3 2 1 6
6 3 2 4 6 8
9 9 4 5 3 2

Ваша матрица после сортировки:
1 2 4 5 7 9
0 1 2 3 5 9
0 0 2 3 5 5
0 0 0 1 2 6
0 0 0 0 6 8
0 0 0 0 0 2

Для завершения нажмите <Enter>Для продолжения нажмите любую клавишу . . .
```

Рисунок 8 – Результаты работы программы

## **Вывод**

В этой лабораторной работе был изучен метод сортировки структур данных; усовершенствованы навыки процедурного программирования на языке C/C++ при решении задач обработки статических и динамических массивов. Эта лабораторная работа показалась мне легче предыдущей, так как она менее объемная и состоит из одной задачи. Единственная сложность, с которой я столкнулась - составить функцию для сортирования массива согласно области, данной в индивидуальном задании.