**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ**



**МОСКОВСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ**

**Факультет Информационных технологий**

***Кафедра Информатики и информационных технологий***

**направление подготовки**

**09.03.02 «Информационные системы и технологии»**

**ЛАБОРАТОРНАЯ РАБОТА № 12-13**

**Дисциплина:** Основы алгоритмизации и программирования

# Тема: "Алгоритм сортировки «Выбором»"

# Выполнил: студент группы 211-721

Дерендяев Дмитрий Сергеевич

(Фамилия И.О.)

**Дата, подпись** 4.12.2021  ***\_\_\_\_\_\_\_\_\_\_\_***

(Дата) (Подпись)

**Проверил: Новичков Иван Константинович *\_\_\_\_\_\_\_\_\_\_\_***

(Фамилия И.О., степень, звание) **(Оценка)**

**Дата, подпись** \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  ***\_\_\_\_\_\_\_\_\_\_\_***

(Дата) (Подпись)

**Замечания: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Москва**

**2021**

# Лабораторная работа №12-13 "Алгоритм сортировки «Выбором»"

**Цель:** Получить практические навыки разработке алгоритмов и их программной реализации.

**Понятие алгоритма:  
Сортировка выбором (*Selection sort*)** — алгоритм сортировки. Может быть как устойчивый, так и неустойчивый. На массиве из *n* элементов имеет время выполнения в худшем, среднем и лучшем случае Θ(*n*2), предполагая, что сравнения делаются за постоянное время

Это возможно, самый простой в реализации алгоритм сортировки. Как и в большинстве других подобных алгоритмов, в его основе лежит операция сравнения. Сравнивая каждый элемент с каждым, и в случае необходимости производя обмен, метод приводит последовательность к необходимому упорядоченному виду.

**Идея алгоритма:**

**Пусть имеется массив A размером N, тогда сортировка выбором сводится к следующему:**

* берем первый элемент последовательности A[i], здесь i – номер элемента, для первого i равен 1;
* находим минимальный (максимальный) элемент последовательности и запоминаем его номер;
* если номер первого элемента и номер найденного элемента не совпадают, тогда два этих элемента обмениваются значениями, иначе никаких манипуляций не происходит;
* увеличиваем i на 1 и продолжаем сортировку оставшейся части массива.

С каждым последующим шагом размер подмассива, с которым работает алгоритм, уменьшается.

**Задачи:**

Необходимо выполнить и оформить описание следующих пунктов:

1. Сформулировать идею алгоритма
2. Выполнить словесное представление алгоритма
3. Выполнить полнить представление алгоритма с помощью блок схем с использованием элемента модификации и без него.
4. Выполнить программную реализацию алгоритмов на языке С с использованием параметрического цикла и цикла с предусловием.

****

**Словесное описание алгоритма:  
n,n2 – длина массива, step – шаг**

1. **Сортировка начинается с первого элемента i=0**
2. **Если i < N - 1, то п. 3, иначе к пункту 12**
3. **min = i, j = i + 1**
4. **Если j < N, то к пункту , иначе к пункту 9**
5. **Ищем локальный минимум. Если array[j] < array[min], то к пункту 6, иначе к пункту 7**
6. **Запоминаем новый индекс (min = j)**
7. **j++**
8. **К пункту 4.**
9. **Обмен значениями mas[i] и mas[min].**
10. **i++**
11. **К пункту 2**
12. **Конец алгоритма**

**Листинг программы:**

#include <iostream>//подключение необходимых библиотек

using namespace std;//определение пространства имен

int main()

{

int N = 7, j, i, buf, min;//подготовка необходимых переменных, в том числе размера массива, параметров циклов, буферных переменных и индекса минимального элемента

int mas[] = { 6, 2, 1, 10, 3, 7, 4 };//объявление массива

for (i = 0; i < N - 1; i++) //начало цикла прохода по всему массиву от первого элемента до последнего с индексом N-1

{

for (j = i + 1, min = i; j < N; j++) //начало вложенного цикла поиска индекса минимального элемента в неотсортированной части массива

{

if (mas[j] < mas[min])//анализ(сравнение) значения текущего элемента со значением минимального элемена неотсортированной части массива

min = j;//присваивание переменной min индекс минимального значения неотсортированной части массива

}

buf = mas[i];//производение обмена значений переменных, используя буферную переменную buf

mas[i] = mas[min];

mas[min] = buf;

}

for (int i = 0; i < N; i++)//выводим массив на экран

{

cout << mas[i] << ' ';

}

cout << endl << endl;

//-------------------------------------------------

int mas2[] = { 6, 5, 1, 15, 3, 2, 4 };//объявление массива

int n, nte, min2, temp, gr = 0;//подготовка необходимых переменных, в том числе размера массива, параметров циклов, буферных переменных и индекса минимального элемента

n = sizeof(mas2) / sizeof(int);//определение размера массива

while (gr < n - 1)//начало цикла прохода по всему массиву от первого элемента до последнего с индексом N-1

{

min2 = gr;//присваивание переменной min индекса первого элемента на данном этапе

nte = gr + 1;//номер текущего элмента равен границе исследуемого массива + 1(для альнейшего анализа)

while (nte < n)//вложенный цикл будет выполняться, пока номер текущего элемента меньше размера самого массива

{

if (mas2[min2] > mas2[nte])//сравнение значений текущего элемента и значения элемента неотсортированной части(увеличение до тех пор, пока не закончится массив)

{

min2 = nte;//рписваивание переменной min2 номер аминимального элемента

}

nte++;//(увеличение номера текущего элемента до тех пор, пока не закончится массив)

}

temp = mas2[gr];//производение обмена значений переменных, используя буферную переменную buf

mas2[gr] = mas2[min2];

mas2[min2] = temp;

gr++;//продвижение по массиву на шаг вперед

}

for (gr = 0; gr < n; gr++)//вывод массива на экран

{

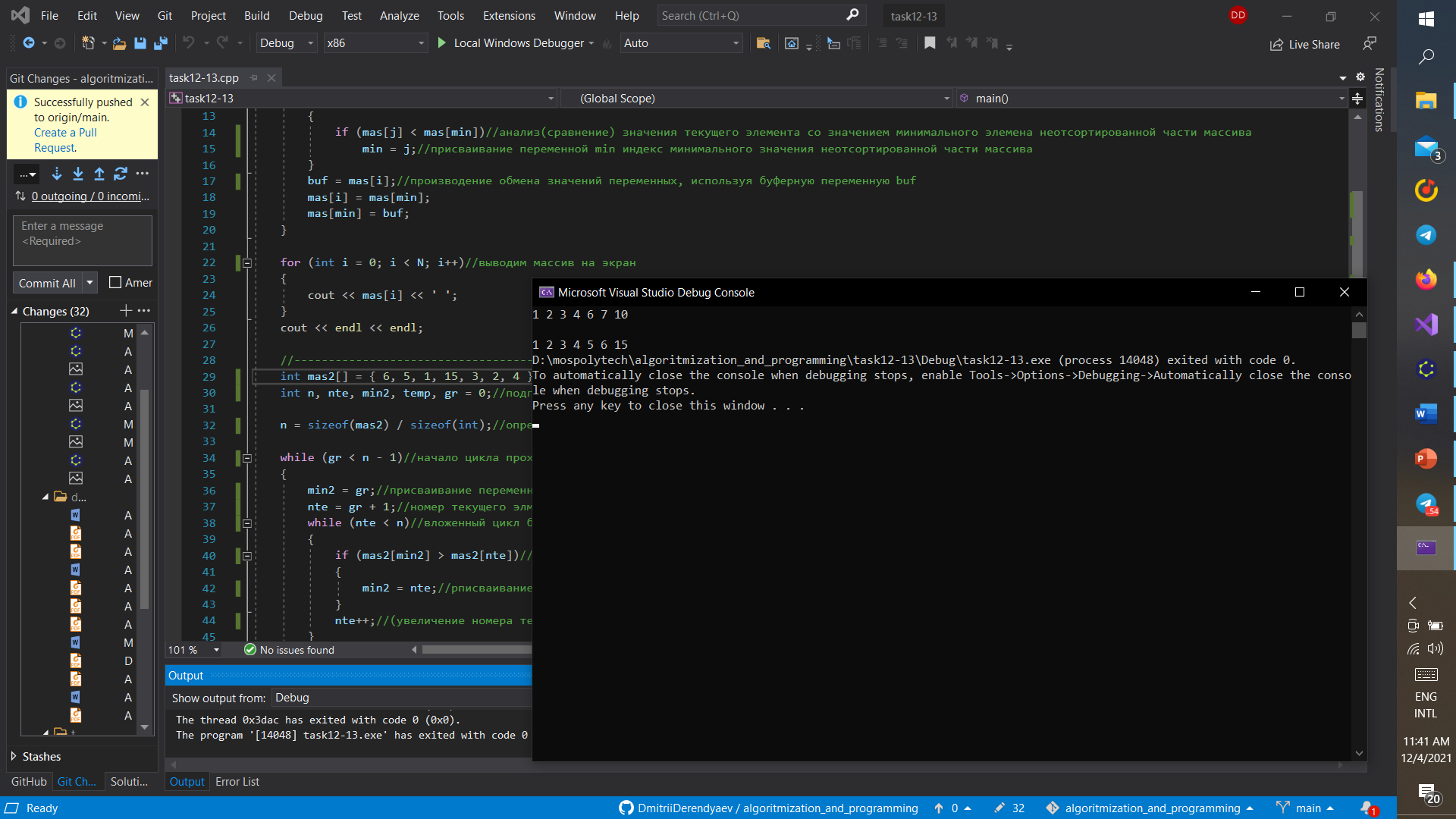
cout << mas2[gr] << ' ';

}

cout << ' ';

return 0;//завершение программы

}



*При необходимости, вы можете найти всю историю разработки программы на моем GitHub:*

[*https://github.com/DmitriiDerendyaev/algoritmization\_and\_programming*](https://github.com/DmitriiDerendyaev/algoritmization_and_programming)