



МОСКОВСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ

Факультет Информационных технологий
Кафедра Информатики и информационных технологий

направление подготовки
09.03.02 «Информационные системы и технологии»

ЛАБОРАТОРНАЯ РАБОТА № 8-9

Дисциплина: Основы алгоритмизации и программирования

Тема: "Алгоритм сортировки «вставками»"

Выполнил: студент группы 211-721

Дерендяев Дмитрий Сергеевич
(Фамилия И.О.)

Дата, подпись 6.11.2021 _____
(Дата) (Подпись)

Проверил: Новичков Иван Константинович _____
(Фамилия И.О., степень, звание) (Оценка)

Дата, подпись _____
(Дата) (Подпись)

Замечания:

Москва

2021

Лабораторная работа №8-9

"Алгоритм сортировки «вставками»"

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

Понятие алгоритма:

Сортировка вставками (англ. *Insertion sort*) — алгоритм сортировки, в котором элементы входной последовательности просматриваются по одному, и каждый новый поступивший элемент размещается в подходящее место среди ранее упорядоченных элементов.

Вычислительная сложность — $O(n^2)$.

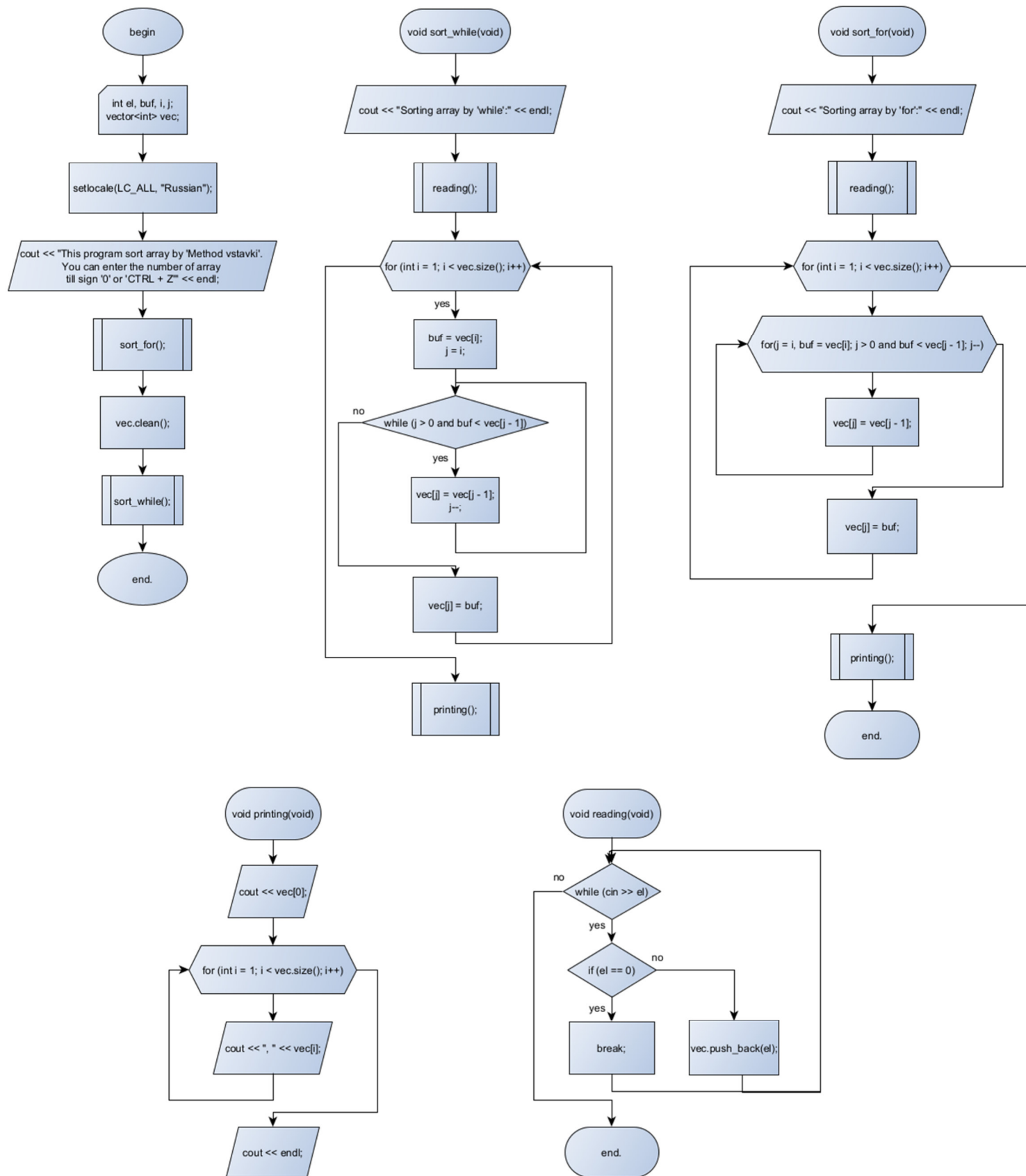
Идея алгоритма:

Сортируемый массив можно разделить на две части — отсортированная часть и неотсортированная. В начале сортировки первый элемент массива считается отсортированным, все остальные — не отсортированные. Начиная со второго элемента массива и заканчивая последним, алгоритм вставляет неотсортированный элемент массива в нужную позицию в отсортированной части массива.

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:

1. Номер анализируемого элемента равен 1
2. Если номер анализируемого элемента меньше размера массива, то п.3; иначе п.9
3. Запоминаем значение анализируемого элемента
4. Номер текущего элемента равен номеру анализируемого элемента
5. Если номер текущего элемента больше «0» и значение анализируемого элемента меньше значения предшествующего текущему, то п.6; иначе п.7
6. Значение текущего элемента равно значению элемента с номером на единицу меньше, п.5
7. В текущий элемент записать значение анализируемого элемента
8. Уменьшить на 1 номер анализируемого элемента, п.2(зависит от реализации)
9. Конец алгоритма

Листинг программы:

```
#include <iostream> //подключение необходимых библиотек
#include <vector>
#include <algorithm>

using namespace std; //определение пространства имен

int el, buf, i, j; //объявление переменных, участвующих в обработке массива
vector<int> vec; //объявление вектора(массива)

void reading(void) //инициализации функции заполнения вектора(с клавиатуры)
{
    while (cin >> el) //начало цикла while и работа его до тех пор, пока вводятся
        значения с клавиатуры(Stop: Ctrl+Z) - НЕ РЕКОМЕНДУЕТСЯ
    {
        if (el == 0) //конец строки для определенной задачи по символу 0
            break; //если был нажат 0, то выход из подпрограммы записи вектора
        else
            vec.push_back(el); //внесение очередного элемента в конец вектора
    }
}

void printing(void) //инициализация функции печати/вывода элементов на экран
{
    cout << vec[0];
    for (int i = 1; i < vec.size(); i++) //цикл вывода элементов на экран с индексом от
        начала до конца вектора(по размеру массива)
    {
        cout << ", " << vec[i]; //печать значения
    }
    cout << endl; //перенос строки
}

void sort_while(void) //инициализация функции сортировки массива с помощью цикла while
{
    cout << "Sorting array by 'while':" << endl; //предупреждение
    reading(); //вызов функции записи вектора
    for (int i = 1; i < vec.size(); i++) //сортировка массива начинается со второго
        элемента
    {
        buf = vec[i]; //помещаем в буферную ячейку значение числа, которое будем в
        дальнейшем анализировать
        j = i; //внутренний цикл начинается с номера анализируемого элемента
```

```

        while (j > 0 and buf < vec[j - 1])//цикл выполняется пока элемент не выходит
за его пределы
            //и он не больше, чем номер анализируемого элемента j и значение
анализируемого элемента меньше, чем предыдущее
            {
                vec[j] = vec[j - 1];//если условие все-таки выполняется то элементы
меняются местами
                j--;//уменьшение на единицу номера анализируемого элемента для того,
чтобы произвести проверку элемента, расположенного ближе к началу
            }
            vec[j] = buf;//перемещение элемента буферной ячейки в ячейку текущего
элемента
        }
        printing();//вывод вектор на экран, используя подпрограмму
    }

void sort_for(void)//инициализация функции печати/вывода элементов на экран
{
    cout << "Sorting array by 'for':" << endl;//предупреждение
    reading();//вызов функции записи вектора
    for (int i = 1; i < vec.size(); i++)//инициализация цикла, который выполняется, пока
не закончатся все элементы массива
    {
        for(j = i, buf = vec[i]; j > 0 and buf < vec[j - 1]; j--)//цикл выполняется
пока элемент не выходит за его пределы
            //и он не больше, чем номер анализируемого элемента j и значение
анализируемого элемента меньше, чем предыдущее
            {
                vec[j] = vec[j - 1];//если условие все-таки выполняется то элементы
меняются местами
            }
            vec[j] = buf;//перемещение элемента буферной ячейки в ячейку текущего
элемента
        }
        printing();//вывод вектор на экран, используя подпрограмму
    }
}

int main();//начало основной программы
{
    setlocale(LC_ALL, "Russian");//установка локали, поддержка Русского языка
    cout << "This program sort array by 'Method vstavki'. You can enter the number of
array till sign '0' or 'CTRL + Z'" << endl;//предупреждение

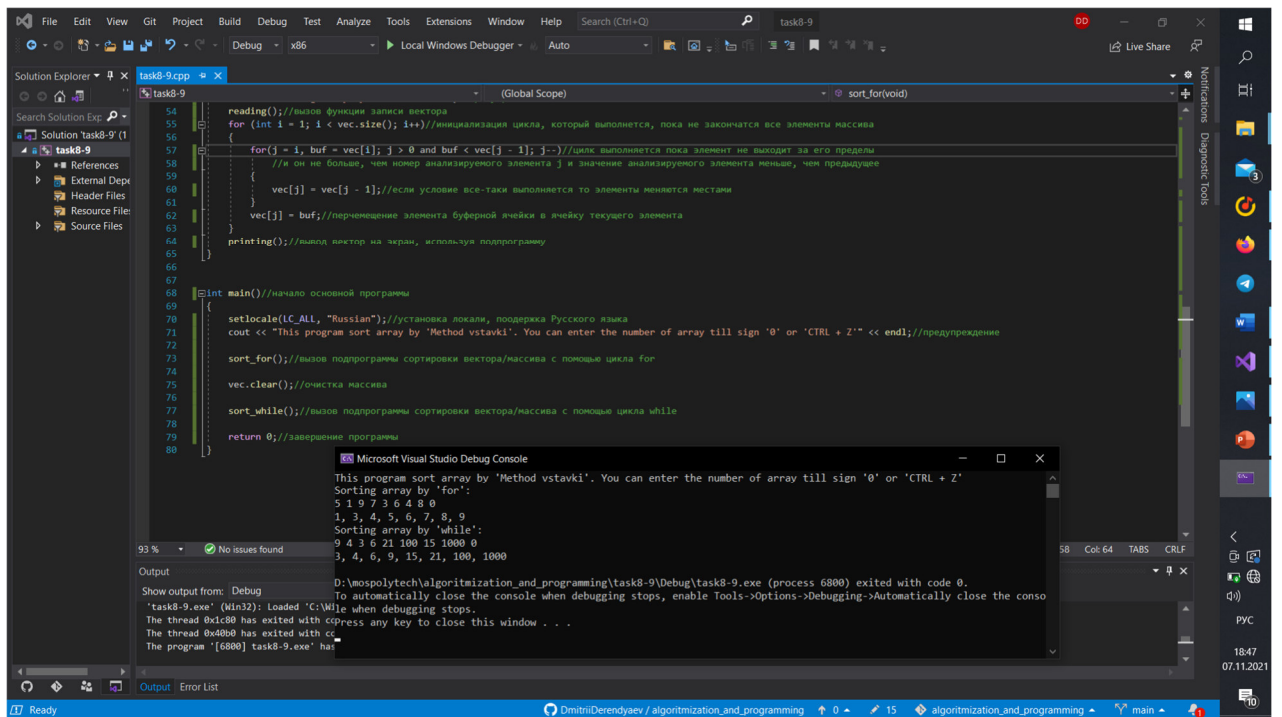
    sort_for();//вызов подпрограммы сортировки вектора/массива с помощью цикла for

    vec.clear();//очистка массива

    sort_while();//вызов подпрограммы сортировки вектора/массива с помощью цикла while

    return 0;//завершение программы
}

```



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

https://github.com/DmitriiDerendyaev/algoritmization_and_programming