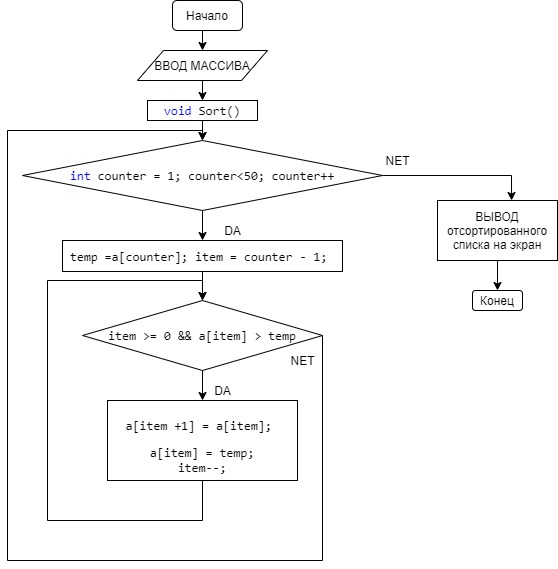
Лабораторная работа №7

«Сортировка»

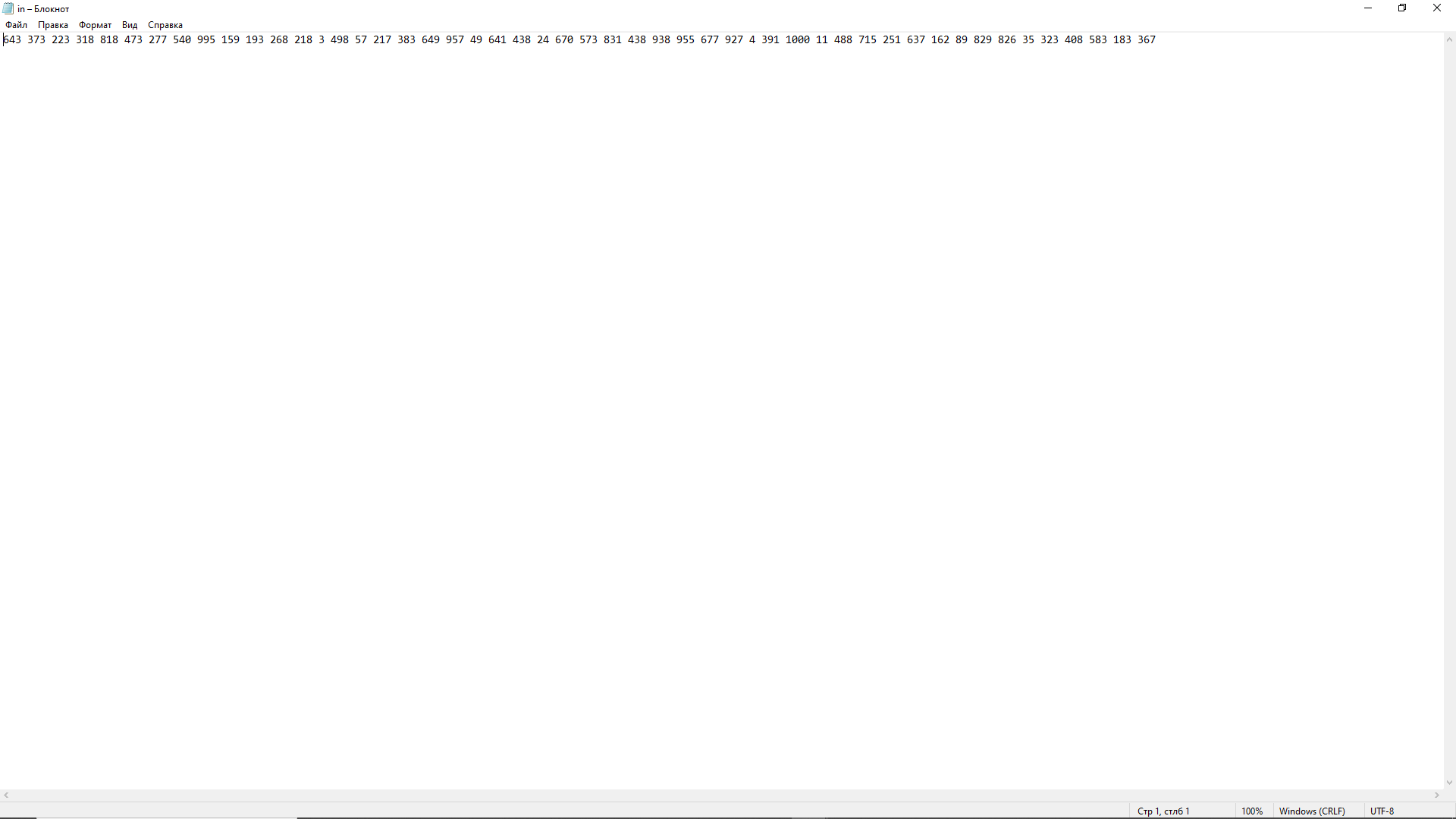
Вариант№2

Задание1- Составить программу сортировки заданной последовательности, используя метод простых вставок. Исходные, промежуточные и итоговые данные хранить в виде соответствующих структур данных (внутренняя сортировка).

Блок-схема:



Входной Файл:



Код:

#include <fstream>

#include <string>

#include<iostream>

#include<cstdlib>

#include<ctime>

using namespace std;

class AR { // Класс иметирующий массив

private:

int\* a; //Указатель

int n; //Количество элементов

public:

AR(int max) // Создаём конструктор

{

a = new int[max]; // Выделение памяти для хранения

n = 0; // Пока элементы отсутствуют

}

void ins(int v) // Функция вставки элементов

{

a[n] = v;

n++; // Увеличиваем

}

void disp() // Функция вывода массива на экран

{

for (int i = 0; i < n; i++)

{

if (i % 20 == 0)

cout << endl;

cout << a[i] << " ";

}

cout << endl;

}

void Sort() // Сортировка методом простых вставок

{

int temp, // временная переменная для хранения значения элемента сортируемого массива

item; // индекс предыдущего элемента

for (int counter = 1; counter < 50; counter++)

{

temp = a[counter]; // инициализируем временную переменную текущим значением элемента массива

item = counter - 1; // запоминаем индекс предыдущего элемента массива

while (item >= 0 && a[item] > temp) // пока индекс не равен 0 и предыдущий элемент массива больше текущего

{

}

}

}

};

int main(int argc, char\*\* argv)

{

setlocale(LC\_ALL, "Russian");

int i;

string path = "in.txt";

ifstream in; // Объявляем переменную (входной поток) и открываем файл

in.open(path);

if (!in.is\_open()) //Проверка наличия файла с элементами

{

cout << "Файл не найден" << endl;

}

else

cout << "Файл с 50 элементами массива открыт\n" << endl;

int ch;

int maxsize = 50; //В массиве 50 элементов

AR Sort(maxsize); //Экземпляр класса

while (!in.eof()) // Построчное считывание

{

in >> ch;

Sort.ins(ch);

}

cout << "Массив до сортировки:";

Sort.disp(); //Вывод массива до сортировки

Sort.Sort(); //Сортируем

cout << endl;

cout << "Сортировка по возрастанию\n" << endl;

cout << "Массив после сортировки:" << endl;

Sort.disp(); //Вывод массива после сортировки

cout << endl;

cout << "Вывод значения параметров эффективности\n" << endl;

const int s = 50;

double time2, c, p; // Так как числа с плавующей точкой

time2 = s \* s; // Оценка сложности по времени сортировки

c = 0.25 \* s \* s; // Для вывода количества сортировок

p = 0.25 \* s \* s; //Для перестановок

cout << "Оценка сложности кода по времени сортировки : " << time2 << " мс\n" << endl; // Уранение для времени

cout << "Количество перестановок: " << p << endl << endl;

cout << "Количество сортировок : " << c << endl << endl;

in.close(); // Очистка

system("pause");

return 0;

}

Cкриншот:

