Министерство науки и высшего образования Российской Федерации

Федеральное государственное автономное образовательное учреждение

высшего образования

**«Пермский национальный исследовательский**

**политехнический университет»**

Электротехнический факультет

Кафедра «Информационные технологии и автоматизированные системы»

Направление подготовки: 09.03.01 - «Информатика и вычислительная техника»

**Лабораторная работа**

**“Методы сортировки массивов”**

Выполнил студент гр. ИВТ-24-2б

Исмагилов Артур Альбертович

Проверил:

Доц. каф. ИТАС

О.А.Полякова.

г.Пермь, 2025

**Метод пузырька**

**Описание**

Алгоритм начинается с конца. Просматриваем 2 соседних элемента и если левый элемент будет больше правого, то меняем их местами. В результате прогона минимальный элемент окажется в начале. Исключаем первый элемент из рассмотрения и попрано сравниваем от последнего до 2 и т.д.Выполняем просмотр n-1 раз каждой пары соседних элеменов, начиная от последнего элемента, равному шагу.

**Код программы**

#include<iostream>

#include<clocale>

using namespace std;

int main()

{

setlocale(LC\_ALL, "RUS");

const int n = 7; int r;

int m[n] = {};

cout << "исходный массив ";

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

{

cin >> m[i];

}

cout << endl;

for (int i = 0; i < n - 1; i++)

{

for (int j = n - 1; j > i; j--)

{

if (m[j] < m[j - 1])

{

r = m[j];

m[j] = m[j - 1];

m[j - 1] = r;

}

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

{

cout << m[i] << " ";

}

cout << endl;

}

}

cout << "отсортированный массив ";

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

{

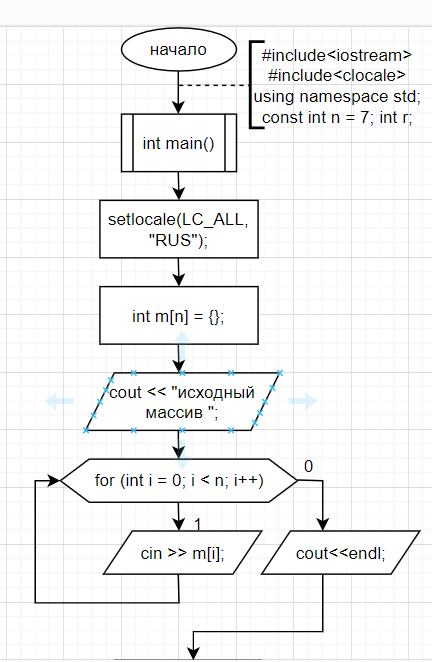
cout << m[i] << " ";

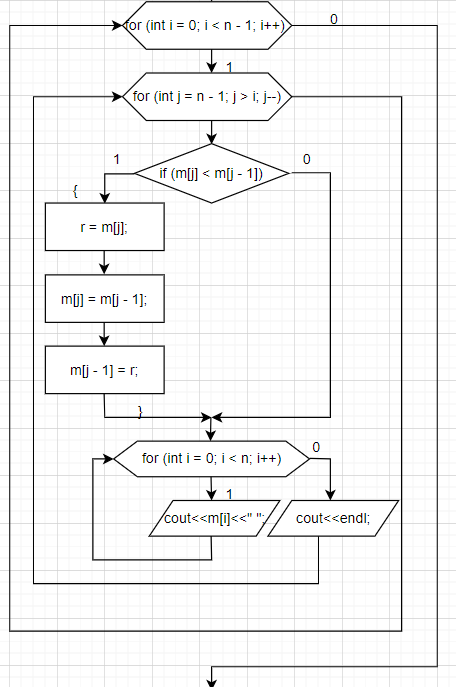
}

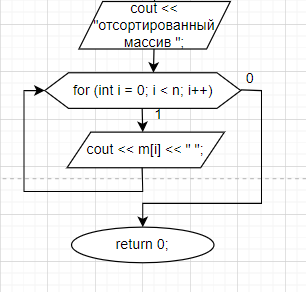
return 0;

}

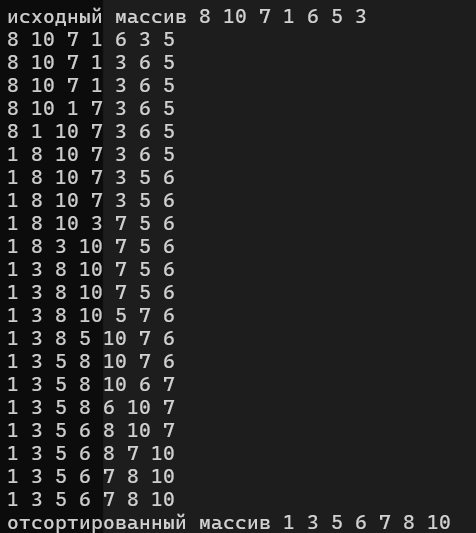
**Блок схема**



****

****

**Вывод программы**



**Метод выбора**

**Описание**

Находим в массиве минимальный элемент и меняем местами с первым элементом. Этот же элемент исключаем из обработки. Находим минимальный элемент из необработанной части и так же меняем его местами со вторым элемнтом и т.д. Выполняем до тех пор пока массив неотсортируется.

**Код программы**

#include<iostream>

#include<clocale>

using namespace std;

int main()

{

setlocale(LC\_ALL, "RUS");

int const n = 7; int r;

int m[n] = {};

cout << "исходный массив ";

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

{

cin>>m[i];

}

cout << endl;

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

{

int min = i;

for (int j = i + 1; j < n; j++)

{

if (m[j] < m[min])

{

min = j;

}

}

r = m[min];

m[min] = m[i];

m[i] = r;

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

{

cout << m[i] << " ";

}

cout << endl;

}

cout << "отсортированыый массив ";

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

{

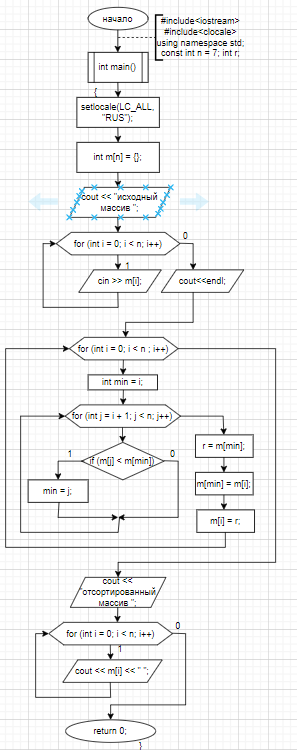
cout << m[i] << " ";

}

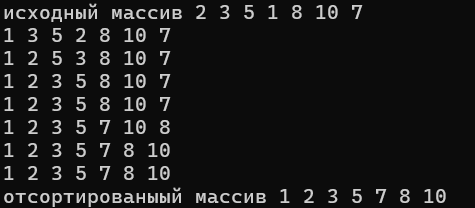
return 0;

}

**Блок схема**

****

**Вывод программы**



**Метод вставки**

**Описание**

На 1 шаге отсортированная часть состоит только из одного первого элемента. Элементы из неотсортированной части поочередно выбираются и вставляются в отсортированную часть так, чтобы упорядоченность отсортировочной части не нарушалась. Таким образом, необходимо сделать n-1 проход. Цикл по перебору элементов неотсортированной части будет внешним. Внутренний цикл перебирает элементы отсортированной части и находит место для элемента, который вставляется в отсортированную часть.

**Код программы**

#include<iostream>

#include<clocale>

using namespace std;

int main()

{

setlocale(LC\_ALL, "RUS");

int const n = 7; int r; int j;

int m[n] = {};

cout << "исходный массив ";

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

{

cin >> m[i];

}

cout << endl;

for (int i = 1; i < n; i++)

{

r = m[i];

j = i - 1;

while (j >= 0 && r < m[j])

{

m[j+1] = m[j];

j--;

}

m[j + 1] = r;

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

{

cout << m[i] << " ";

}

cout << endl;

}

cout << "отсортированный массив ";

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

{

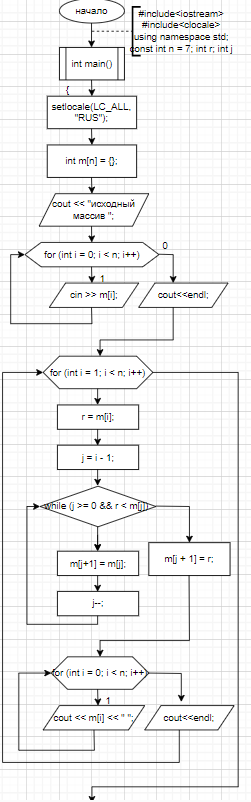
cout << m[i] << " ";

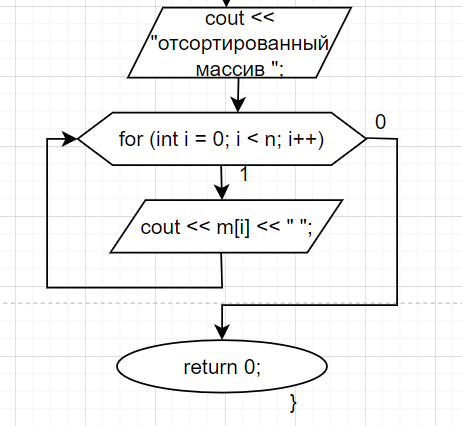
}

return 0;

}

**Блок схема**

****

****

**Вывод программы**

