Пермский Национальный Исследовательский Политехнический Университет

Электротехнический факультет

Кафедра «Информационные технологии и автоматизированные системы»

I семестр

«Основы алгоритмизации и программирования»

Лабораторная работа № 4

Тема: «Сортировка массивов»

Выполнил:

студент группы РИС-20-2б

Пономарев Егор Витальевич

Проверила:

доцент кафедры ИТАС

Полякова О. А.

Пермь, 2020

**Постановка задачи**

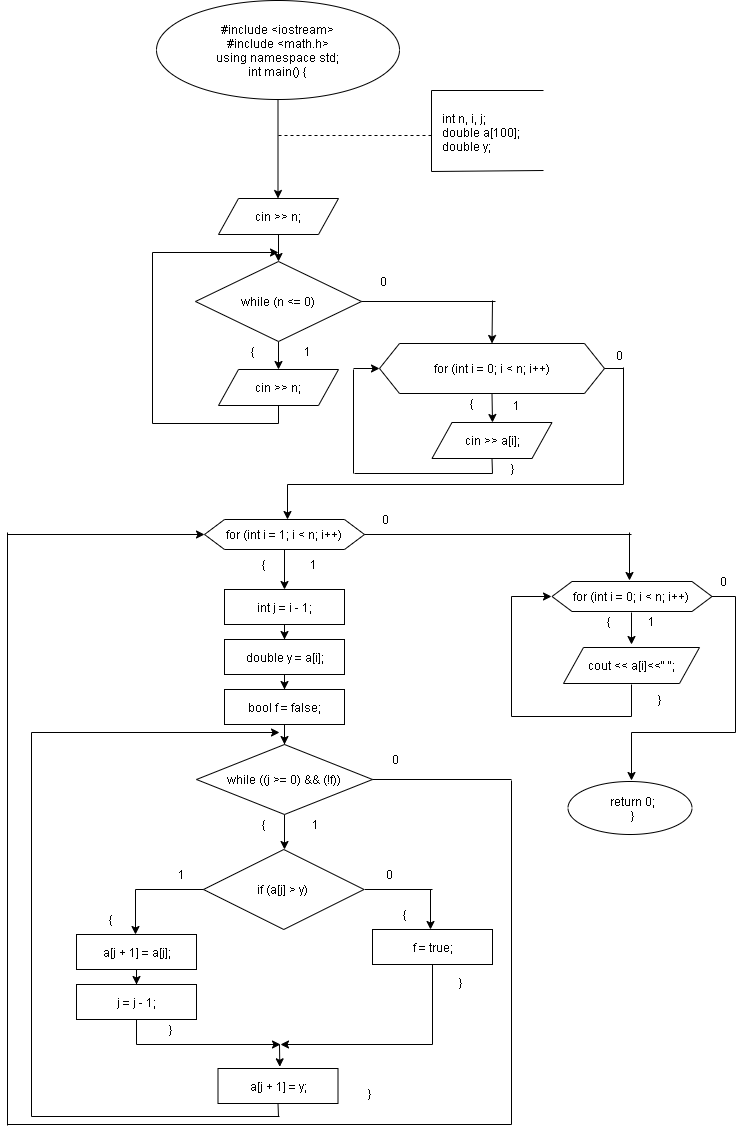
Отсортировать массив по возрастанию тремя методами: метод вставки, метод выбора, метод обмена (пузырька).

**Анализ задачи**

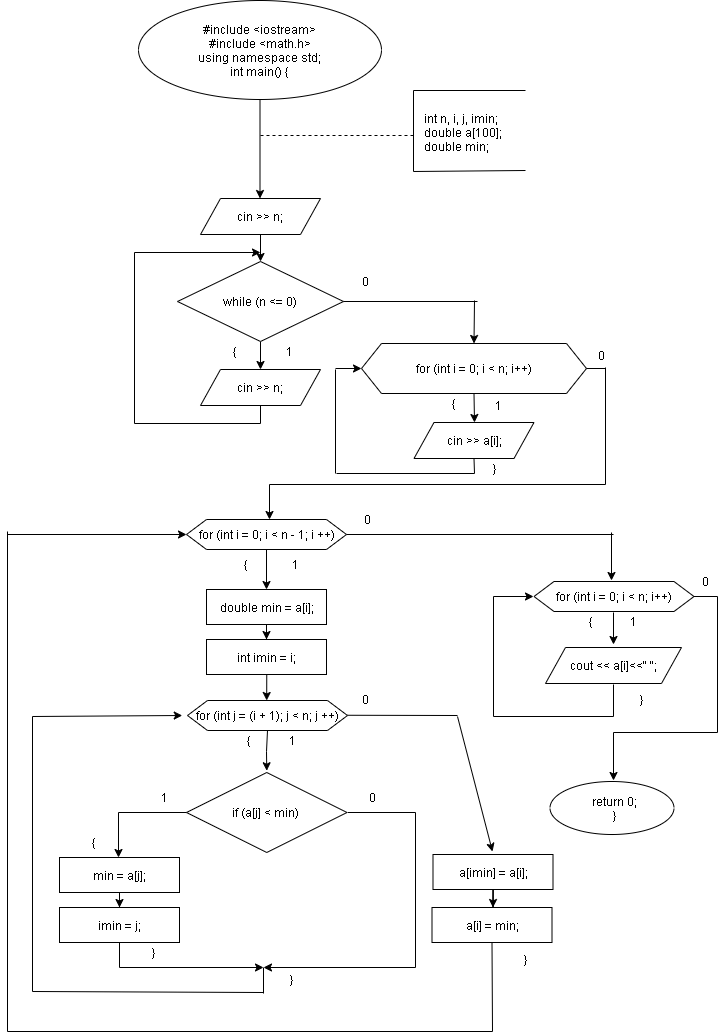
1. **Метод вставки**
   1. Массив делится на 2 части: отсортированную и неотсортированную
   2. Элементы неотсортированной части поочерёдно выбираются и вставляются в отсортированную часть так, чтобы упорядоченность отсортированной части не нарушалась
   3. Сначала отсортированная часть содержит 1 элемент, остальные в неотсортированной части (то есть алгоритм состоит из (n-1) проходов)
2. **Метод выбора**
3. Находим в массиве минимальный элемент и меняем местами его с первым
4. Первый элемент исключаем из обработки
5. Находим минимальный элемент среди оставшихся и меняем его местами со вторым
6. Продолжаем повторять поиск минимального и менять местами
7. **Метод обмена (пузырька)**
8. Просматриваем два соседних элемента с конца
9. Если левый элемент > правого элемента, то меняем их местами
10. В результате минимальный элемент встаёт на первое место (всплыл первый пузырёк)
11. По такому же принципу получаем второй отсортированный элемент
12. Выполняем просмотр элементов n-1 раз.

**Блок-схемы**

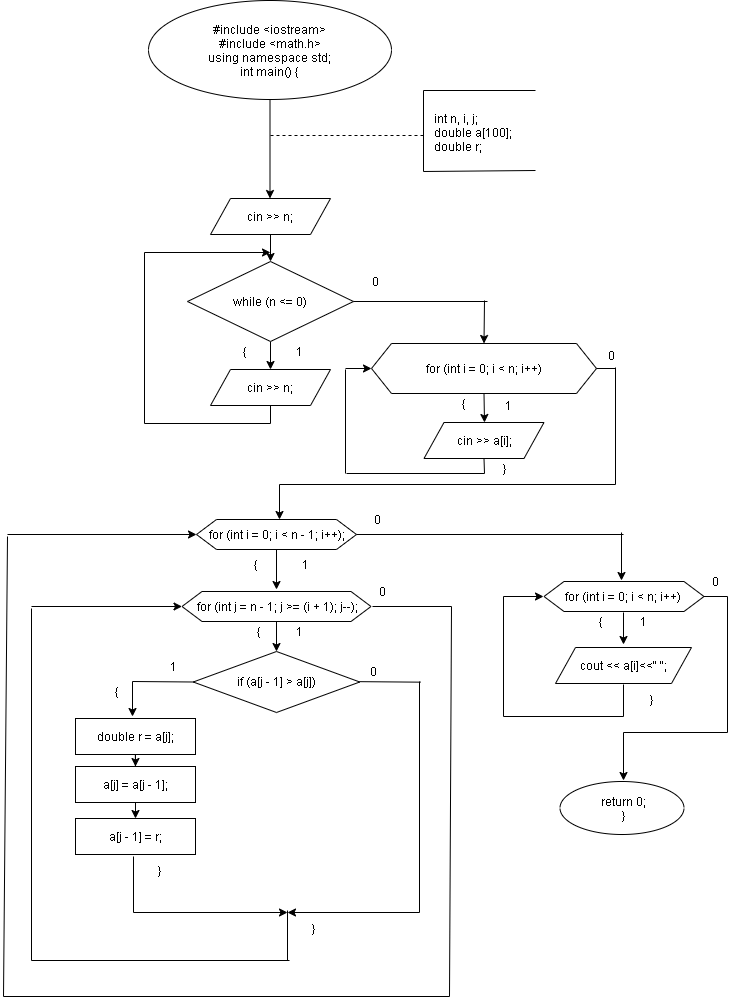
* 1. **Метод вставки**

****

1. **Метод выбора**

****

1. **Метод обмена**

****

**Код C++**

**1. Метод вставки**

#include <iostream>

#include <math.h>

using namespace std;

int main() {

int n;

cin >> n;

while (n <= 0) { // проверка на дурака

cin >> n;

}

double a[100];

for (int i = 0; i < n; i++) { // инициализируем массив

cin >> a[i];

}

for (int i = 1; i < n; i++) { // проходим по массиву, начиная со 2 элемента

int j = i - 1; // присваиваем j номер предыдущего элемента перед a[i]

double y = a[i]; // присваиваем y сравниваемый элемент массива

bool f = false; // инициализируем флажок

while ((j >= 0) && (!f)) {

if (a[j] > y) { // проверяем условие

a[j + 1] = a[j]; // перемещаем элемент массива на 1 вправо

j = j - 1; // переходим к предыдущему элементу

}

else {

f = true; // меняем значение флажка, т.к. нашли нужный элемент массива

}

a[j + 1] = y; // ставим элемент массива в нужное место

}

}

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

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

}

return 0;

}

1. **Метод выбора**

#include <iostream>

#include <math.h>

using namespace std;

int main() {

int n;

cin >> n;

while (n <= 0) { // проверка на дурака

cin >> n;

}

double a[100];

for (int i = 0; i < n; i++) { // инициализируем массив

cin >> a[i];

}

for (int i = 0; i < n - 1; i++) { // проходим по массиву n - 1 раз

double min = a[i]; // переменная для хранения минимального значения

int imin = i; // переменная для хранения индекса мин. значения

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

if (a[j] < min) { // находим мин. элемент

min = a[j];

imin = j;

}

}

a[imin] = a[i]; // сортируем массив

a[i] = min;

}

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

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

}

return 0;

}

1. **Метод обмена**

#include <iostream>

#include <math.h>

using namespace std;

int main() {

int n;

cin >> n;

while (n <= 0) { // проверка на дурака

cin >> n;

}

double a[100];

for (int i = 0; i < n; i++) { // инициализируем массив

cin >> a[i];

}

for (int i = 0; i < n - 1; i++) { // проходим по массиву n-1 раз, нач. со 2 элемента

for (int j = n - 1; j >= (i + 1); j--) { // проходим по массиву, в обр. сторону

if (a[j - 1] > a[j]) { // сравниваем 2 сосед. элемента и меняем их местами

double r = a[j];

a[j] = a[j - 1];

a[j - 1] = r;

}

}

}

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

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

}

return 0;

}

**Результат выполнения**

**1. Метод вставки**

****

****

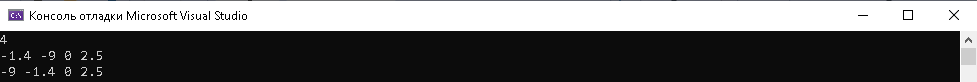
**2. Метод выбора**

****

****

1. **Метод обмена (пузырька)**

****

****