**Министерство образования и науки Российской Федерации  
Федеральное государственное бюджетное образовательное учреждение  
высшего профессионального образования  
«Московский государственный технический университет имени Н.Э. Баумана»  
(МГТУ им. Н.Э. Баумана)**

|  |  |
| --- | --- |
| **ФАКУЛЬТЕТ** | **«Информатики и систем управления»** |
| **КАФЕДРА** | **5-13** |

**ОТЧЕТ**

***ПО ЛАБОРАТОРНОЙ РАБОТЕ № 6***

ПО ДИСЦИПЛИНЕ:«Основы программирования»

Студент Гаврилов Л.Я.

(Группа) (Подпись, дата) (И.О. Фамилия)

Проверил Козлов А.Д.

(Подпись, дата) (И.О. Фамилия)

*2018  г.*

# Задания на лабораторную работу

## Задание 1

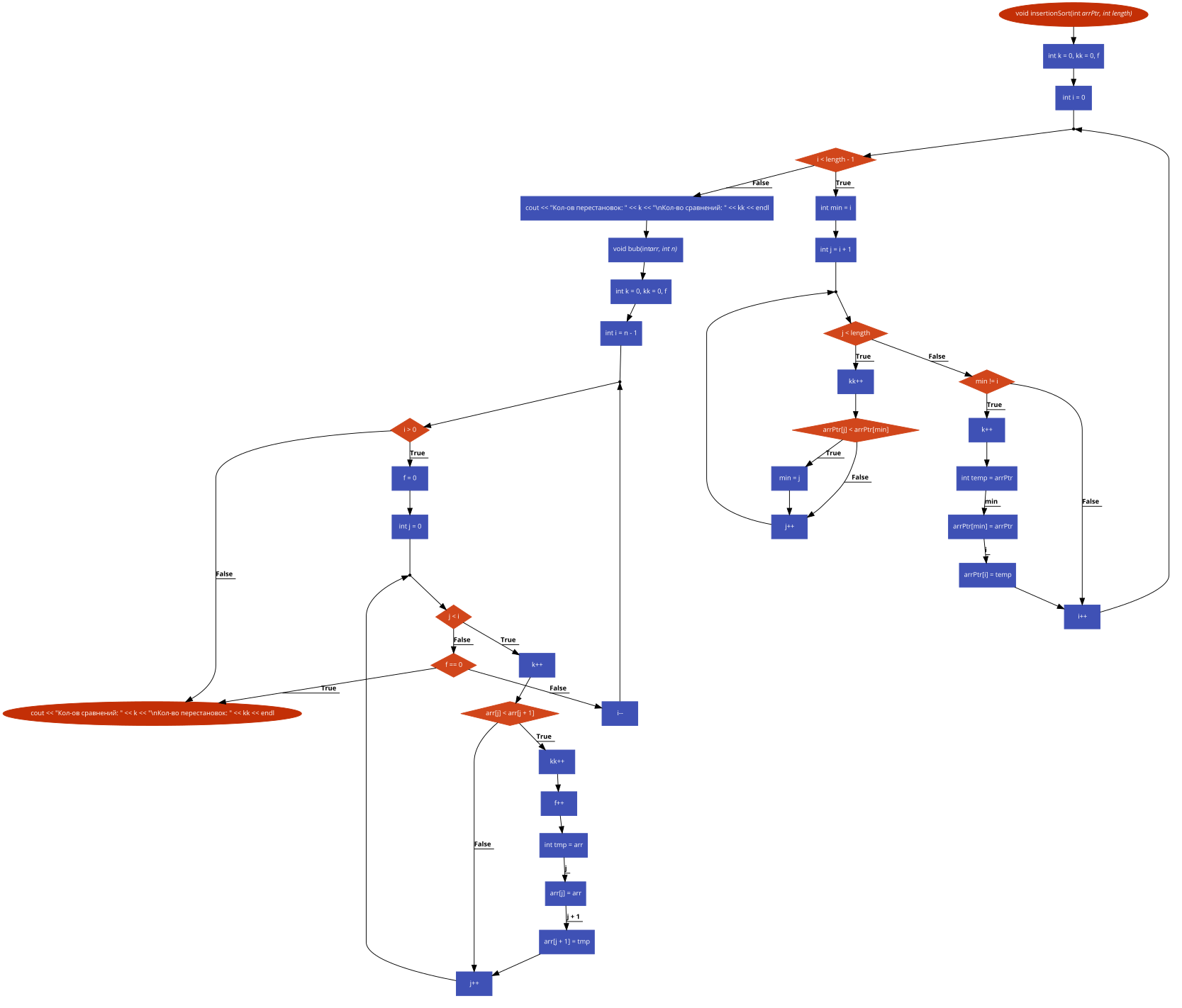
Отсортировать числовой массив n методом выбора максимального (минимального) элемента и методом пузырькового всплытия. По окончании сортировки вывести отсортированный массив и количество сделанных сравнений и перестановок элементов. Умножить на -1 и отсортировать.

# Описание разработанных программ

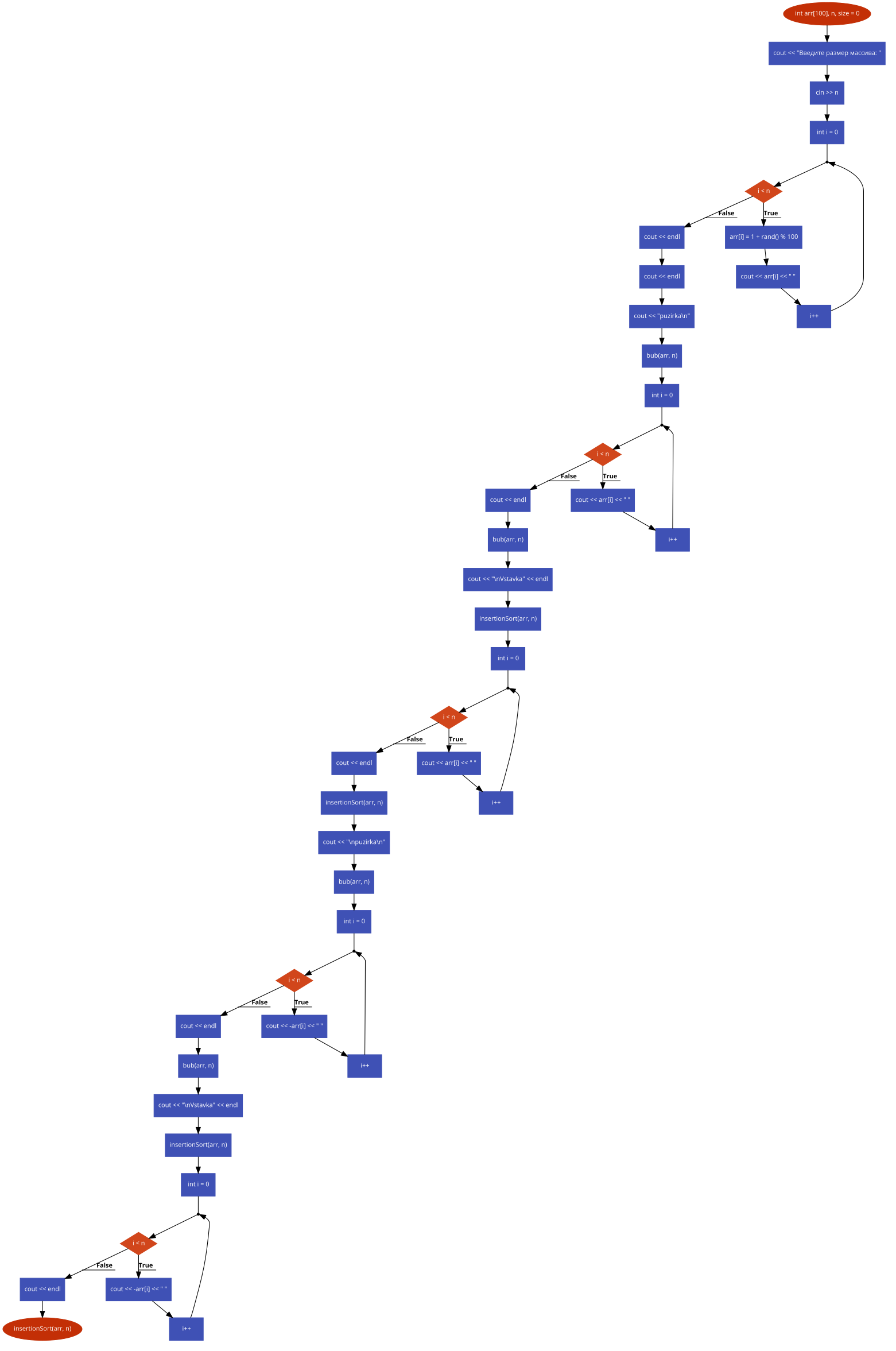
## Задание 1

### Описание используемых алгоритмов

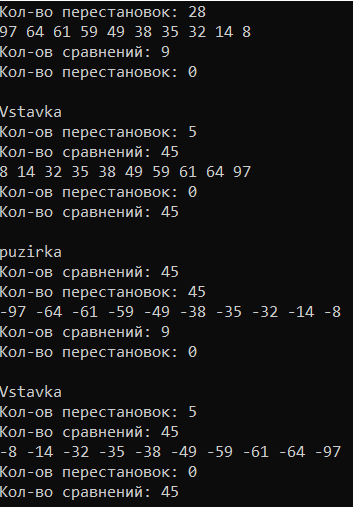
Алгоритм для функций сортировки, основная суть программы.

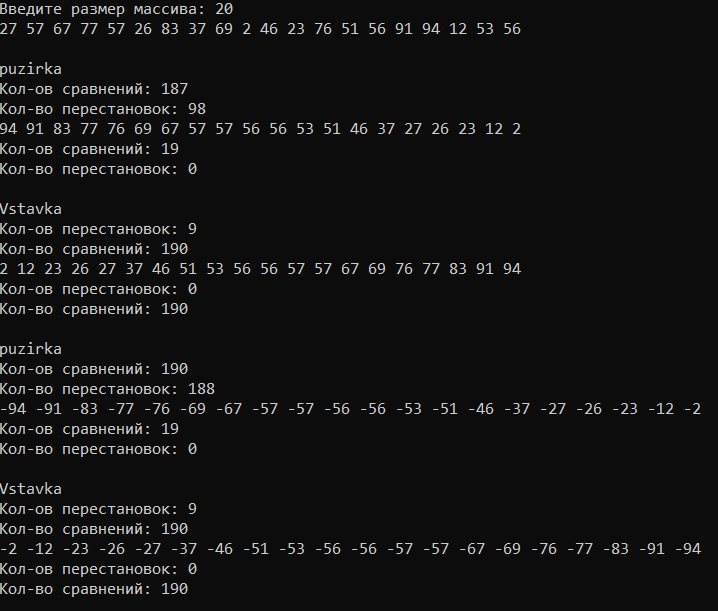


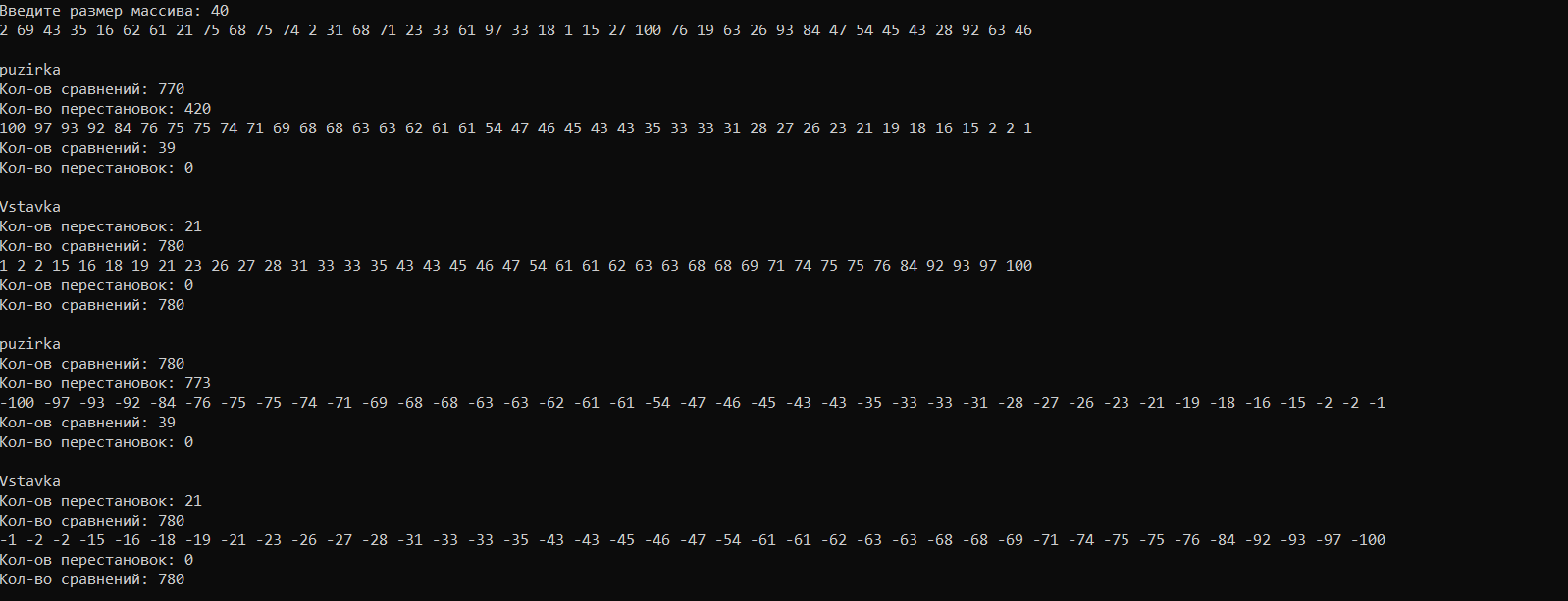
Функция main()



### Пример выполнения программы







### Текст программы

#include "pch.h"

#include <iostream>

#include <ctime>

using namespace std;

void insertionSort(int \*arrPtr, int length);

void bub(int\* arr, int n);

int main()

{

srand(time(NULL));

setlocale(LC\_ALL, "Ru");

int arr[100], n, size = 0;

cout << "Введите размер массива: ";

cin >> n;

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

arr[i] = 1 + rand() % 100;

cout << arr[i] << " ";

}

cout << endl;

cout << endl;

cout << "puzirka\n";

bub(arr, n);

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

cout << arr[i] << " ";

}

cout << endl;

bub(arr, n);

cout << "\nVstavka" << endl;

insertionSort(arr, n);

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

cout << arr[i] << " ";

}

cout << endl;

insertionSort(arr, n);

cout << "\npuzirka\n";

bub(arr, n);

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

cout << -arr[i] << " ";

}

cout << endl;

bub(arr, n);

cout << "\nVstavka" << endl;

insertionSort(arr, n);

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

cout << -arr[i] << " ";

}

cout << endl;

insertionSort(arr, n);

return 0;

}

void insertionSort(int \*arrPtr, int length)

{

int k = 0, kk = 0, f;

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

{

int min = i;

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

kk++;

if (arrPtr[j] < arrPtr[min]) {

min = j;

}

}

if (min != i) {

k++;

int temp = arrPtr[min];

arrPtr[min] = arrPtr[i];

arrPtr[i] = temp;

}

}

cout << "Кол-ов перестановок: " << k << "\nКол-во сравнений: " << kk << endl;

}

void bub(int\* arr, int n) {

int k = 0, kk = 0, f;

for (int i = n - 1; i > 0; i--)

{

f = 0;

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

k++;

if (arr[j] < arr[j + 1])

{

kk++;

f++;

int tmp = arr[j];

arr[j] = arr[j + 1];

arr[j + 1] = tmp;

}

}

if (f == 0) break;

}

cout << "Кол-ов сравнений: " << k << "\nКол-во перестановок: " << kk << endl;

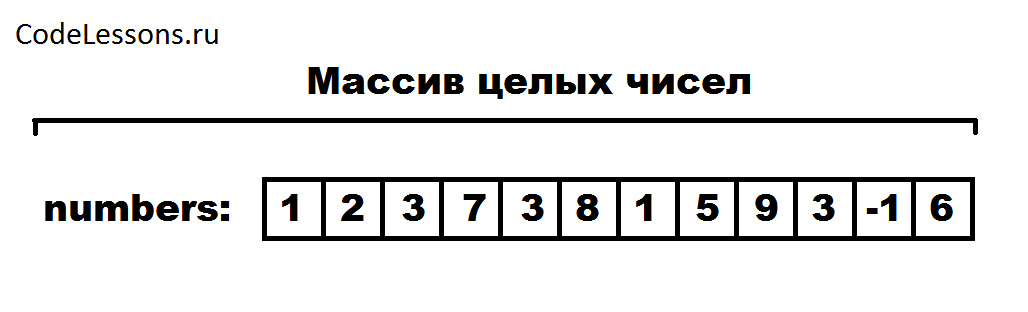
}

# Выводы

## Что такое массив?

Массив — это набор однотипных данных. Например, вы можете хранить все носки в одном месте. Для этого вы используете полку. В C++ множество элементов хранятся в массивах.

Для лучшего понимания давайте рассмотрим следующею картинку:



По ней мы можем понять следующее:

* Каждый массив должен иметь свое название.
* Он может в себе содержать от одного элемента до бесконечности (это в теории, на практике размер массива ограничивается памятью компьютера).
* Все элементы должны быть одного типа. Так, например, вы не можете в одном массиве хранить переменные типа int и типа double.

### Инициализация массива

Для того чтобы использовать массивы в своих программах, его надо сначала объявить. Для этого нужно использовать следующею конструкцию:

<тип> <имя массива> [ <кол-во элементов > ];

double arr[100];

***Указатель на функцию***

Указатель на функцию (function pointer) хранит адрес функции. По сути указатель на функцию содержит адрес первого байта в памяти, по которому располагается выполняемый код функции.

Самым распространенным указателем на функцию является ее имя. С помощью имени функции можно вызывать ее и получать результат ее работы.

тип (\*имя\_массива[размер]) (параметры)

void bub(int\* arr, int n)