1. Пока элементы слева от запомненного значения больше чем запомненное – перемещаем их на позицию вправо. Получается, что предыдущий элемент займет ячейку запомненного. А тот, что стоит перед предыдущим – переместится в свою очередь на место предыдущего. И так  элементы будут перемещаться друг за другом.
2. По убыванию

#include <iostream>

using namespace std;

int length = 0;

void input(int\* arr) {//ввод

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

cin >> arr[i];

}

}

void InsertionSort(int\* arr) {//сортировка вставками

int key, i;

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

key = arr[j];

i = j - 1;

while (arr[i] <= key && i >= 0) {

arr[i + 1] = arr[i];

--i;

}

arr[i + 1] = key;

}

}

void output(int\* arr) {//вывод

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

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

}

}

int main()

{

int size;

cin >> size;

int\* arr = new int[size];

length = size;

input(arr);

InsertionSort(arr);

output(arr);

return 0;

}

#include <iostream>

using namespace std;

int length = 0;

void input(int\* arr) {//ввод

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

cin >> arr[i];

}

}

void InsertionSort(int\* arr) {//сортировка вставками

int key, i;

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

key = arr[j];

i = j - 1;

while (arr[i] <= key && i >= 0) {

arr[i + 1] = arr[i];

--i;

}

arr[i + 1] = key;

}

}

void LinearSearch(int\* arr, int x) {

int c = 0;

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

if (x == arr[i]) {

c++;

}

}

if (c > 0) {

cout << "true" << endl;

cout << "kolichestvo x: " << c;

}

else {

cout << "false" << endl;

}

}

void output(int\* arr) {//вывод

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

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

}

cout << endl;

}

int main()

{

int size;

int x;

cin >> size;

cout << "x= ";

cin >> x;

int\* arr = new int[size];

length = size;

input(arr);

InsertionSort(arr);

output(arr);

LinearSearch(arr, x);

return 0;

}

1 3 5 4 2