Дз на 14.03.2023 г.

1. наивная сортировка

#include <iostream>

#include <string>

#include <vector>

#include <algorithm>

using namespace std;

int main() {

int n;

cin >> n;

int m = n;

int\* a = new int[n];

int\* b = new int[m];

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

cin >> a[i];

}

int minim = 1000000;

int ind = 0;

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

int h = 0;

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

if (a[j] < a[0]) {

swap(a[j], a[0]);

}

}

b[i] = a[0];

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

a[j] = a[j + 1];

}

n--;

}

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

cout << b[j] << " ";

}

}

2. обратная наивная сортировка

#include <iostream>

#include <string>

#include <vector>

#include <algorithm>

using namespace std;

int main() {

int n;

cin >> n;

int m = n;

int\* a = new int[n];

int\* b = new int[m];

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

cin >> a[i];

}

int minim = 1000000;

int ind = 0;

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

int h = 0;

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

if (a[j] > a[0]) {

swap(a[j], a[0]);

}

}

b[i] = a[0];

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

a[j] = a[j + 1];

}

n--;

}

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

cout << b[j] << " ";

}

}

3. шаблонная функция возвращающая отсортированный массив

#include <iostream>

#include <string>

#include <vector>

#include <algorithm>

using namespace std;

template <typename X>

void printin(X\* a, int n) {

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

cin >> a[i];

}

}

template <typename T>

void sort(T\* a, int n) {

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

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

if (a[j] < a[j - 1]) {

int temp = a[j];

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

a[j - 1] = temp;

}

}

}

}

int main() {

int n;

cin >> n;

int\* a = new int[n];

printin(a, n);

sort(a, n);

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

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

}

}