1. **Алгоритмы поиска в массиве**
   1. **Бинарный поиск**

#include <iostream>

#include <Windows.h>

using namespace std;

/// <summary>

/// Находит индекс числа num в отсортированном массиве mass с помощью бинарного поиска

/// </summary>

/// <param name="mass">Набор чисел</param>

/// <param name="num">Искомое числа в наборе чисел mass</param>

/// <param name="length">Размер массива mass</param>

int BinarySearch(int\* mass, int num, int length)

{

int left = 0,

right = length - 1;

//пока не вышли за пределы массива

do

{

int mid = (right + left) / 2;//находим средний индекс с заданном интервале [left, right]

if (mass[mid] == num)

return mid;

else if (mass[mid] > num)

right = mid - 1;

else if (mass[mid] < num)

left = mid + 1;

} while (left <= right);

return -1;//указываем, что элемент не найден

}

int main()

{

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

const int N = 3;

int a[N] = { 1, 2, 3 };

cout << BinarySearch(a, 3, N) << endl;//2

cout << BinarySearch(a, 2, N) << endl;//1

cout << BinarySearch(a, 1, N) << endl;//0

cout << BinarySearch(a, 11, N) << endl;//-1

return 0;

}

* 1. **Быстрая сортировка**

#include <iostream>

#include <Windows.h>

using namespace std;

/// <summary>

/// Сортирует массив mass с помощью алгортима быстрой сортировки

/// </summary>

/// <param name="mass">Массив из целых чисел</param>

/// <param name="left">Индекс левой границе сортирвоки массива</param>

/// <param name="right">Индекс правой границе сортирвоки массива</param>

void QuickSort(int\* mass, int left, int right)

{

int item = mass[(left + right)/2];//текущий сортируемый элемент (может быть любой, не оязательно брать по середке)

int i = left,

j = right;

do {

//Находим индекс первого элемента, который будет больше item

//в пределах интервала [left, right]

while (mass[i] < item) i++;

//Находим индекс первого элемента, который будет меньше item

//в пределах интервала [left, right]

while (mass[j] > item) j--;

//Если элемент, который больше item расположен

//левее элемента, который меньше item

if (i < j) {

//Меняем эти элементы местами

int copy = mass[i];

mass[i] = mass[j];

mass[j] = copy;

}

else {

i++;

j--;

}

} while (i < j);//повторяем цикл до тех пор, пока оба указателя (i и j) не перескочат через друг друга

if (left < j) QuickSort(mass, left, j);

if (i < right) QuickSort(mass, i, right);

}

int main()

{

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

const int N = 3;

int a[N] = { 3, 1, 4 };

QuickSort(a, 0, N-1);

return 0;

}