

Министерство науки и высшего образования Российской Федерации
Федеральное государственное автономное образовательное учреждение
высшего образования

**Пермский национальный исследовательский
политехнический университет**

Факультет электротехнический

Кафедра ИТАС

ОТЧЁТ

о лабораторной работе №16

Выполнил:
Студент группы ИВТ-23-1Б
Пискунов Д. А.

Проверил:
Доцент кафедры ИТАС
Яруллин Д.В.

Пермь 2024

Задача:

Написать программу на выполняющую три вида поиска: линейный, бинарный, интерполяционный.

Текст программы

```
1  #include <iostream>
2  #include <ctime>
3
4  using namespace std;
5
6  int partition(int* arr, int left, int right) {
7      int pivot = arr[left];
8      int i = left - 1, j = right + 1;
9      while (1) {
10         do {
11             i++;
12         } while (arr[i] < pivot);
13
14         do {
15             j--;
16         } while (arr[j] > pivot);
17
18         if (i >= j) {
19             return j;
20         }
21         swap(arr[i], arr[j]);
22     }
23 }
24
25 void sort(int* arr, int left, int right) {
26     if (left < right) {
27         int pi = partition(arr, left, right);
28         sort(arr, left, pi);
29         sort(arr, pi + 1, right);
30     }
31 }
32
33 void linear_search(int* arr, int size, int key) {
34     int* found = new int[0];
35     int found_size = 0;
36     for (int i = 0; i < size; i++) {
37         if (arr[i] == key) {
38             found[found_size++] = i;
39         }
40     }
41     if (found_size != 0) {
42         cout << "Ключ найден на позиции(ях) ";
43         for (int i = 0; i < found_size; i++) {
44             cout << found[i] + 1 << " ";
45         }
46     }
47     else {
48         cout << "Ключ не найден";
49     }
50     delete[] found;
51 }
52
53 void interpolation_search(int* arr, int size, int key) {
54     if (arr[0] == key) {
55         cout << "\nКлюч найден на позиции 1";
```

```

55     cout << "\nКлюч найден на позиции 1";
56 }
57
58 int start = 0, end = size - 1;
59 int mid, found = -1;
60 while (arr[start] < key and arr[end] >= key) {
61     mid = start + ((key - arr[start]) * (end - start)) / (arr[end] - arr[start]);
62     if (arr[mid] == key) {
63         found = mid;
64         break;
65     }
66     else if (arr[mid] < key) {
67         start = mid + 1;
68     }
69     else if (arr[mid] > key) {
70         end = mid - 1;
71     }
72 }
73 if (found != -1) {
74     cout << "\nКлюч найден на позиции " << found + 1;
75 }
76 else {
77     cout << "\nКлюч не найден";
78 }
79 }
80
81 void binary_search(int* arr, int size, int key) {
82     int start = 0;
83     int end = size - 1;
84
85     while (start <= end) {
86         int mid = start + (end - start) / 2;
87
88         if (arr[mid] == key) {
89             cout << "Ключ найден на позиции " << mid + 1;
90             return;
91         }
92         else if (arr[mid] < key) {
93             start = mid + 1;
94         }
95         else {
96             end = mid - 1;
97         }
98     }
99     cout << "Ключ не найден";
100 }
101
102 void menu(int* arr, bool s, int size) {
103     int choice, pos, key;
104     cout << "\n\nВведите искомое значение\n";
105     cin >> key;
106     cout << "\n\nВыберите метод поиска: \n";
107     cout << "1. Линейный поиск\n";
108     cout << "2. Интерполяционный поиск\n";
109     cout << "3. Бинарный поиск\n";

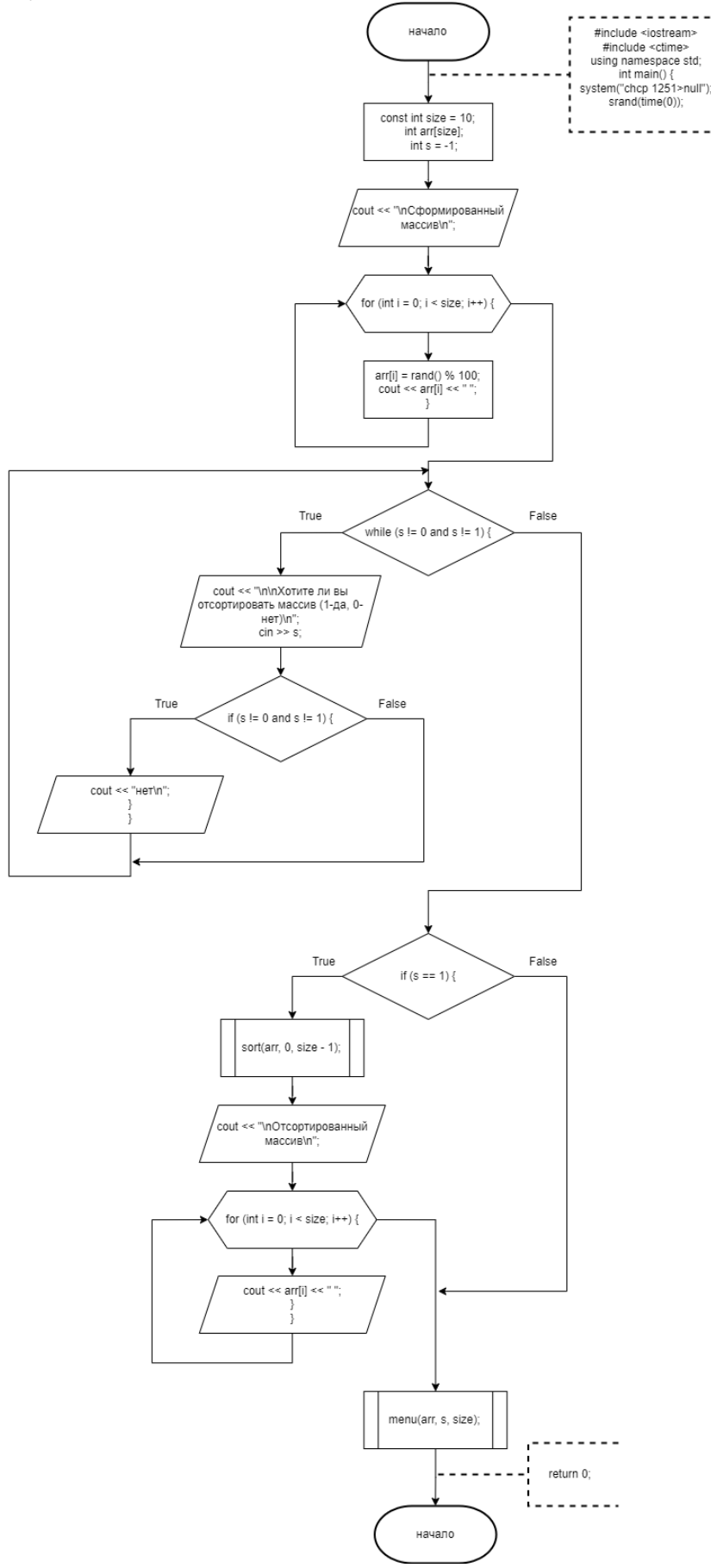
```

```

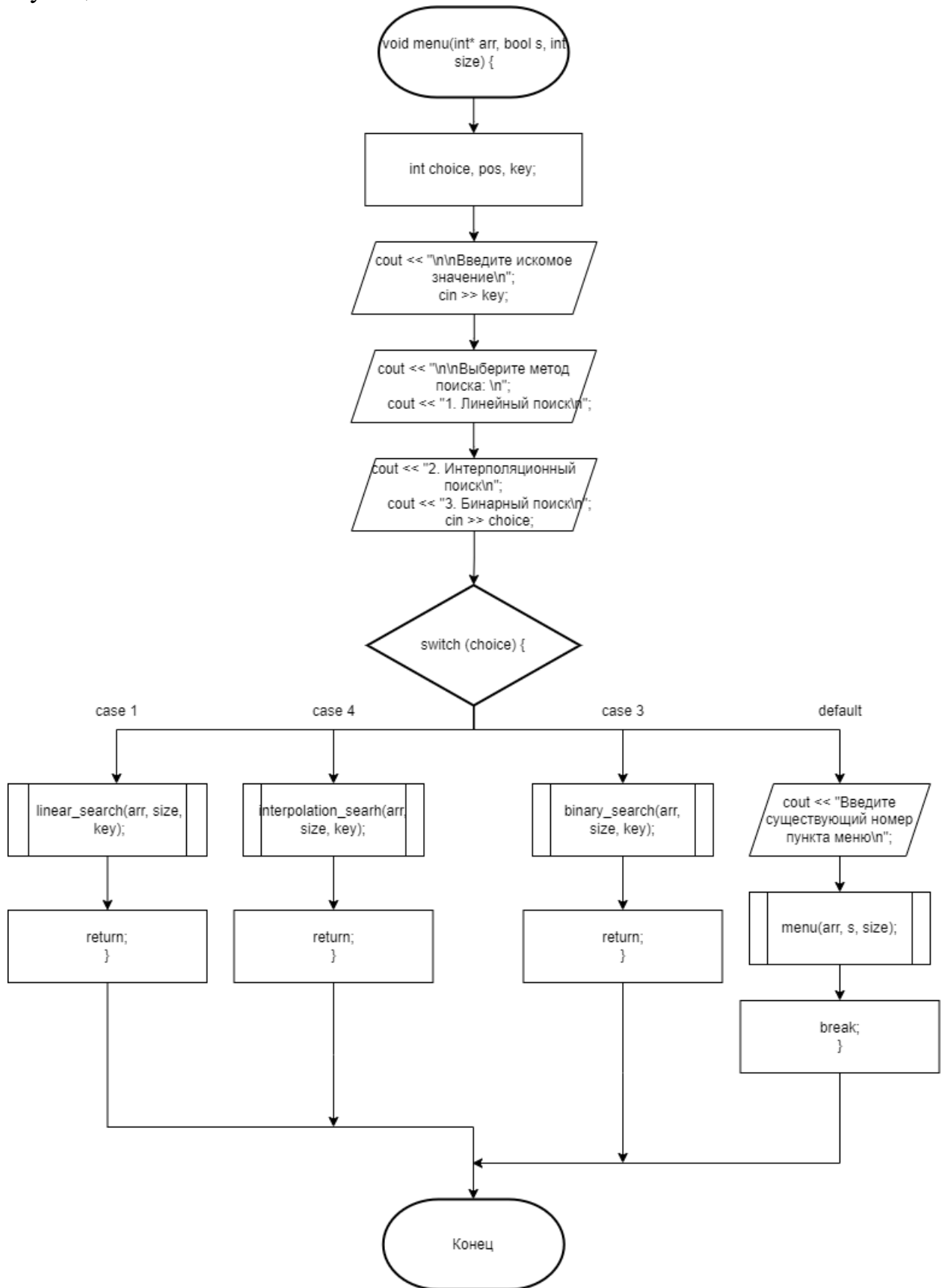
109     cout << "3. Бинарный поиск\n";
110     cin >> choice;
111
112
113     switch (choice) {
114     case 1: {
115         linear_search(arr, size, key);
116         return;
117     }
118     case 2: {
119         interpolation_search(arr, size, key);
120         return;
121     }
122     case 3: {
123         binary_search(arr, size, key);
124         return;
125     }
126     default: {
127         cout << "Введите существующий номер пункта меню\n";
128         menu(arr, s, size);
129         break;
130     }
131     }
132 }
133
134
135 int main() {
136     system("chcp 1251>null");
137     srand(time(0));
138     const int size = 10;
139     int arr[size];
140     int s = -1;
141
142     cout << "\nСформированный массив\n";
143     for (int i = 0; i < size; i++) {
144         arr[i] = rand() % 100;
145         cout << arr[i] << " ";
146     }
147     while (s != 0 and s != 1) {
148         cout << "\n\nХотите ли вы отсортировать массив (1-да, 0-нет)\n";
149         cin >> s;
150         if (s != 0 and s != 1) {
151             cout << "нет\n";
152         }
153     }
154     if (s == 1) {
155         sort(arr, 0, size - 1);
156         cout << "\nОтсортированный массив\n";
157         for (int i = 0; i < size; i++) {
158             cout << arr[i] << " ";
159         }
160     }
161     menu(arr, s, size);
162     return 0;
163 }

```

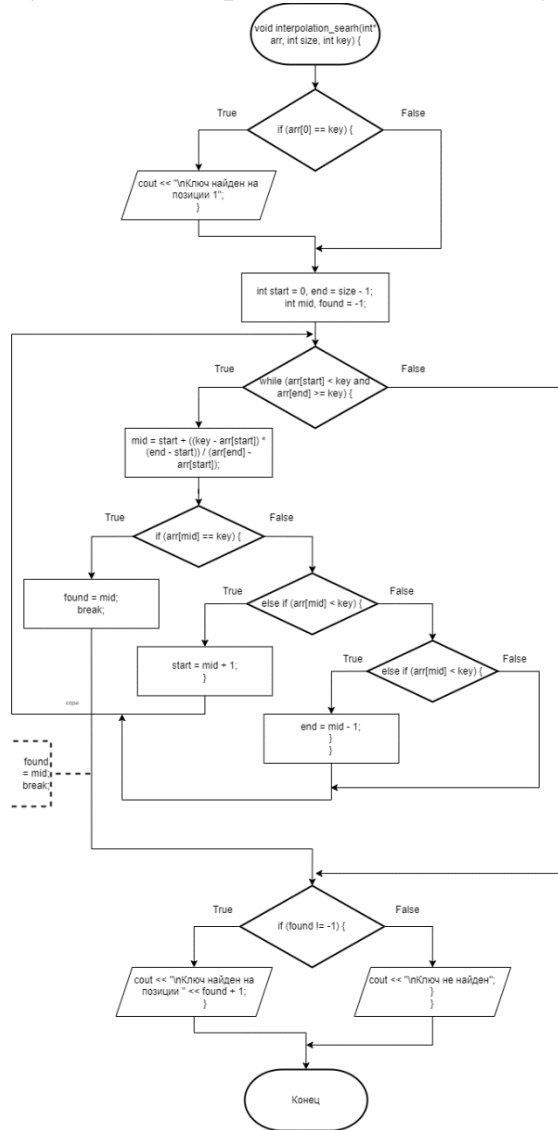
Блок схема
Функция main



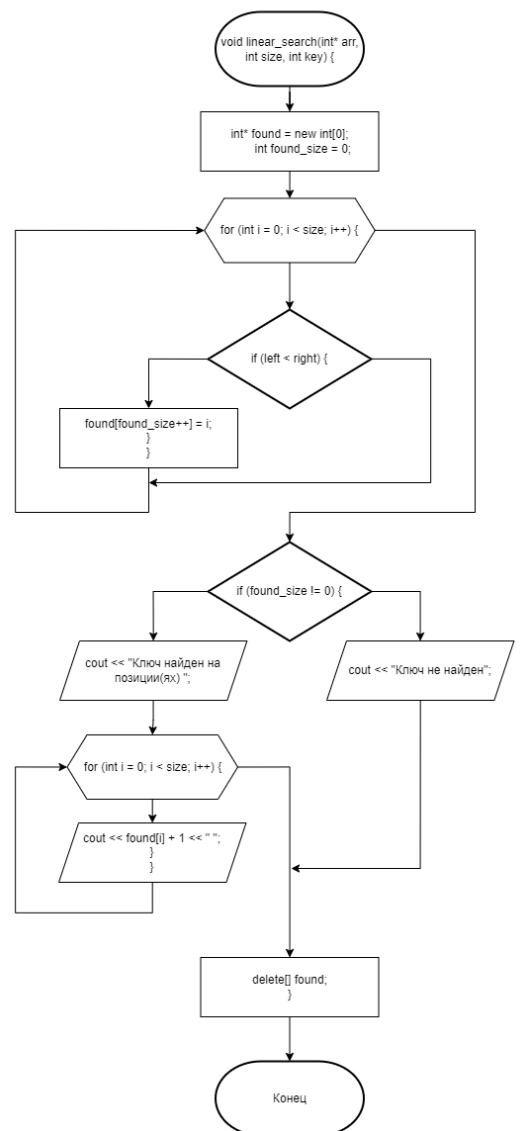
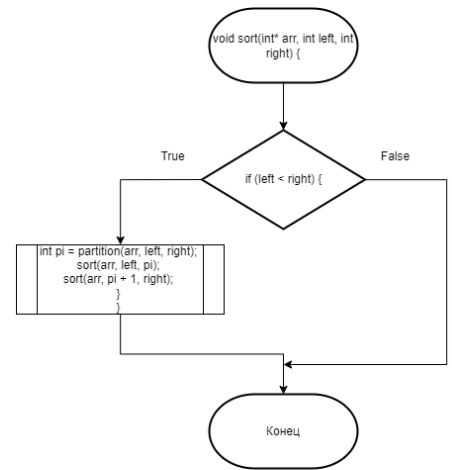
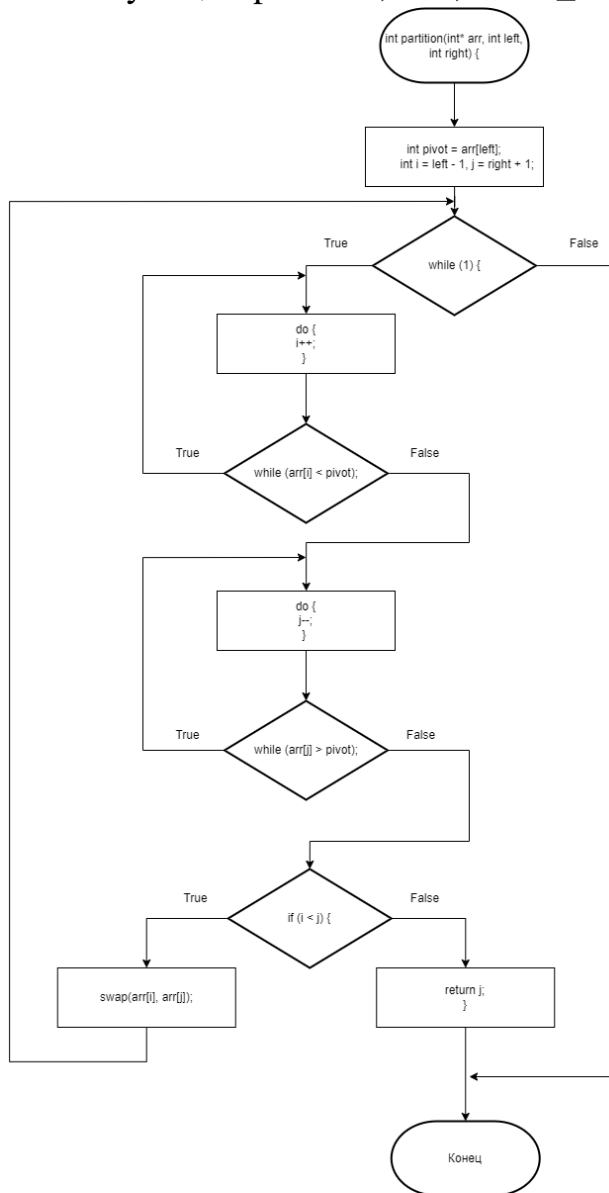
Функция menu



Функции interpolation_search, binary_search



Функции partition, sort, linear_search



Тесты

Сформированный массив
2 76 57 52 67 61 52 20 28 68

Хотите ли вы отсортировать массив (1-да, 0-нет)
0

Введите искомое значение
52

Выберите метод поиска:
1. Линейный поиск
2. Интерполяционный поиск
3. Бинарный поиск
1

Ключ найден на позиции(ях) 4 7

C:\Users\MOKASiH\Desktop\Test\x64\Debug\Test.exe (процесс 884) завершил работу с кодом 0.
Нажмите любую клавишу, чтобы закрыть это окно...

Сформированный массив
92 12 36 67 60 92 16 37 0 46

Хотите ли вы отсортировать массив (1-да, 0-нет)
1

Отсортированный массив
0 12 16 36 37 46 60 67 92 92

Введите искомое значение
36

Выберите метод поиска:
1. Линейный поиск
2. Интерполяционный поиск
3. Бинарный поиск
2

Ключ найден на позиции 4

C:\Users\MOKASiH\Desktop\Test\x64\Debug\Test.exe (процесс 3504) завершил работу с кодом 0.
Нажмите любую клавишу, чтобы закрыть это окно...

Сформированный массив
49 47 48 26 40 67 73 0 24 51

Хотите ли вы отсортировать массив (1-да, 0-нет)
1

Отсортированный массив
0 24 26 40 47 48 49 51 67 73

Введите искомое значение
40

Выберите метод поиска:
1. Линейный поиск
2. Интерполяционный поиск
3. Бинарный поиск
3

Ключ найден на позиции 4

C:\Users\MOKASiH\Desktop\Test\x64\Debug\Test.exe (процесс 10436) завершил работу с кодом 0.
Нажмите любую клавишу, чтобы закрыть это окно...

