

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

**«Пермский национальный исследовательский
политехнический университет»**

Электротехнический факультет
Кафедра «Информационные технологии и автоматизированные системы»
направление подготовки: 09.03.01 – «Информатика и вычислительная
техника»

**Отчет по
лабораторной работе 16**

Выполнил студент гр. ИВТ-23-16
Давыдов Андрей Юрьевич

Проверил:

ст. преп. каф. ИТАС

Яруллин Денис Владимирович

(оценка)

(подпись)

(дата)

г. Пермь, 2023

Постановка задачи

Поиски: линейный, интерполяционный, бинарный

Код программы:

```
#include <iostream>
using namespace std;

// Функция для обмена двух элементов
void swap(int* a, int* b) {
    int t = *a;
    *a = *b;
    *b = t;
}

// Функция для разделения массива (поиск опорного элемента)
int partition(int arr[], int low, int high) {
    int pivot = arr[high]; // опорный элемент
    int i = (low - 1);

    for (int j = low; j <= high - 1; j++) {
        // Если текущий элемент меньше или равен опорному
        if (arr[j] <= pivot) {
            i++; // инкремент индекса меньшего элемента
            swap(&arr[i], &arr[j]);
        }
    }
    swap(&arr[i + 1], &arr[high]);
    return (i + 1);
}

// Функция для выполнения Quick Sort
void quickSort(int arr[], int low, int high) {
    if (low < high) {
        // pi - индекс разделения, arr[pi] теперь на правильном месте
        int pi = partition(arr, low, high);

        // Рекурсивно сортируем элементы до и после разделения
        quickSort(arr, low, pi - 1);
        quickSort(arr, pi + 1, high);
    }
}

// Функция для вывода массива
void printArray(int arr[], int size) {
    for (int i = 0; i < size; i++)
        cout << arr[i] << " ";
    cout << endl;
}

int binarySearch(int arr[], int l, int r, int x) {
    while (l <= r) {
        int m = l + (r - l) / 2;

        // Проверяем, находится ли x в середине
        if (arr[m] == x)
            return m;

        // Если x больше, игнорируем левую половину
        if (arr[m] < x)
            l = m + 1;

        // Если x меньше, игнорируем правую половину
        else
            r = m - 1;
    }

    // Элемент не присутствует в массиве
    return -1;
}

// Линейный поиск
int linearSearch(int arr[], int n, int x) {
    for (int i = 0; i < n; i++) {
        if (arr[i] == x)
            return i;
    }
    return -1;
}
```

```

// Интерполяционный поиск
int interpolationSearch(int arr[], int lo, int hi, int x) {
    int pos;
    if (lo <= hi && x >= arr[lo] && x <= arr[hi]) {
        pos = lo + (((double)(hi - lo) / (arr[hi] - arr[lo])) * (x - arr[lo]));

        if (arr[pos] == x)
            return pos;

        if (arr[pos] < x)
            return interpolationSearch(arr, pos + 1, hi, x);

        if (arr[pos] > x)
            return interpolationSearch(arr, lo, pos - 1, x);
    }
    return -1;
}

// Функция для выбора вида поиска
void searchMenu(int arr[], int n) {
    int choice, x, result;
    cout << "Выберите метод поиска:\n";
    cout << "1. Бинарный поиск\n";
    cout << "2. Линейный поиск\n";
    cout << "3. Интерполяционный поиск\n";
    cout << "Введите номер метода: ";
    cin >> choice;
    cout << "Введите искомое число: ";
    cin >> x;

    switch (choice) {
        case 1:
            result = binarySearch(arr, 0, n - 1, x);
            break;
        case 2:
            result = linearSearch(arr, n, x);
            break;
        case 3:
            result = interpolationSearch(arr, 0, n - 1, x);
            break;
        default:
            cout << "Неверный выбор.\n";
            return;
    }

    if (result != -1)
        cout << "Элемент найден на позиции: " << result + 1 << endl;
    else
        cout << "Элемент не найден.\n";
}

```

```

int main() {
    system("chcp 1251>null");
    int arr[] = { 10, 7, 8, 9, 1, 5 };
    int n = sizeof(arr) / sizeof(arr[0]);
    quickSort(arr, 0, n - 1);
    cout << "Отсортированный массив: \n";
    printArray(arr, n);
    searchMenu(arr, n);
    return 0;
}

```

Тест программы

```
Отсортированный массив:  
1 5 7 8 9 10  
Выберите метод поиска:  
1. Бинарный поиск  
2. Линейный поиск  
3. Интерполяционный поиск  
Введите номер метода: 1  
Введите искомое число: 8  
Элемент найден на позиции: 4
```

```
Отсортированный массив:  
1 5 7 8 9 10  
Выберите метод поиска:  
1. Бинарный поиск  
2. Линейный поиск  
3. Интерполяционный поиск  
Введите номер метода: 2  
Введите искомое число: 5  
Элемент найден на позиции: 2
```

```
Отсортированный массив:  
1 5 7 8 9 10  
Выберите метод поиска:  
1. Бинарный поиск  
2. Линейный поиск  
3. Интерполяционный поиск  
Введите номер метода: 3  
Введите искомое число: 10  
Элемент найден на позиции: 6
```