**ФЕДЕРАЛЬНОЕ АГЕНТСТВО СВЯЗИ**

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ**

**«САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ТЕЛЕКОММУНИКАЦИЙ ИМ. ПРОФ. М.А. БОНЧ-БРУЕВИЧА»**

**(СПбГУТ)**

**Кафедра безопасности информационных систем**

**ОТЧЁТ**

по практической работе работе №5 на тему:   
**«Методы поиска элемента в массиве»**

по дисциплине «Алгоритмы и структуры данных»

Выполнил: студент группы ИСТ-114,Константинов А.Д.

«29» октября 2022 г. \_\_\_\_\_\_\_\_\_\_\_/А.Д. Константинов/

Принял: к.ф.-м.н., доцент, И.А. Моисеев

« » ноября 2022 г. \_\_\_\_\_\_\_\_\_\_\_/ И.А. Моисеев /

**Содержание:**

[Цель работы: 3](#_Toc114692552)

[Листинг кода: 3](#_Toc114692553)

[Результат работы программы: 8](#_Toc114692554)

[Вывод: 8](#_Toc114692555)

[Приложение: 22](#_Toc114692556)

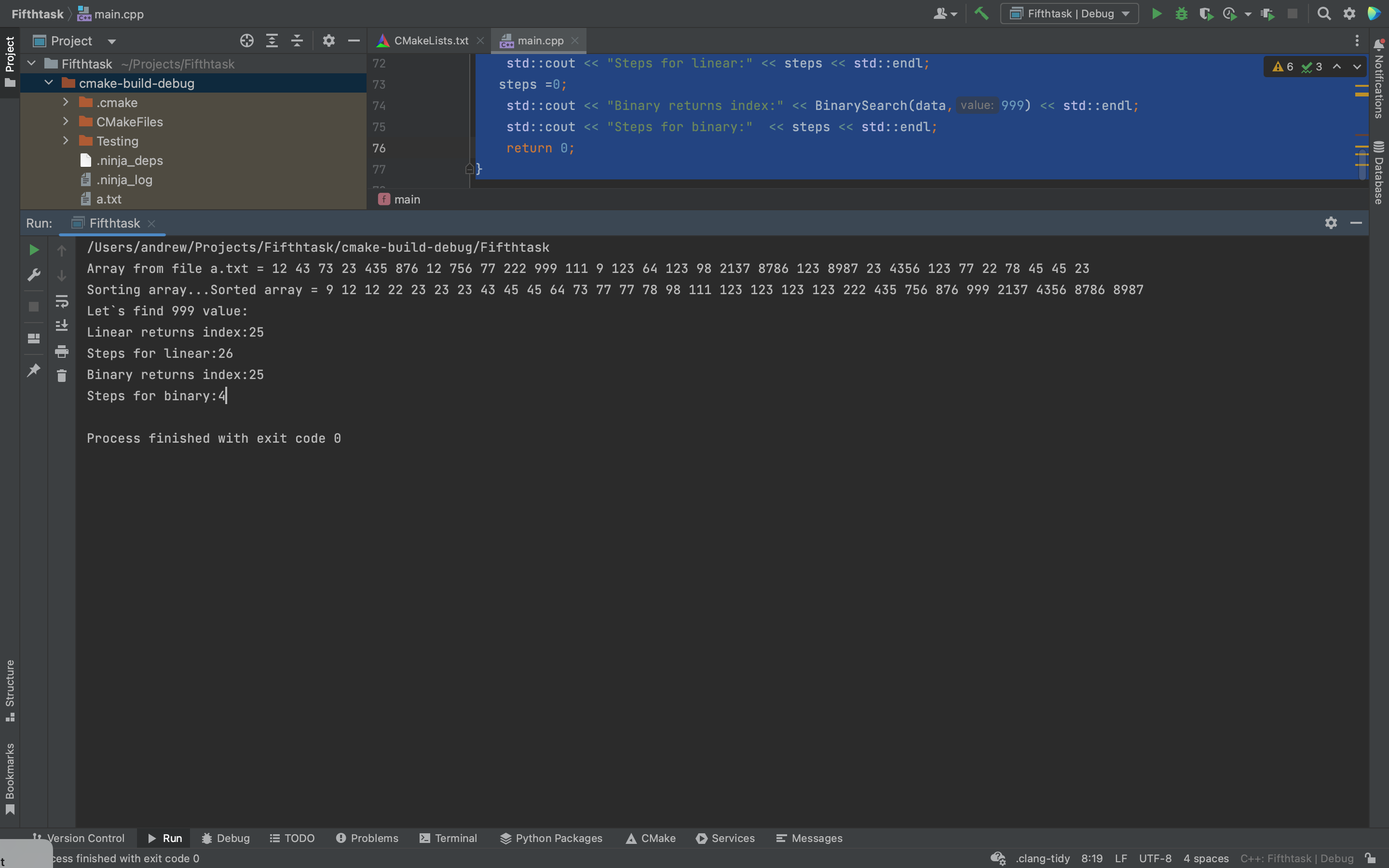
Цель работы:

# В массиве рассмотренном в предыдущей задаче 3 найти заданный элемент методами линейного и бинарного поиска. Оценить их трудоемкость O(n) по методике, изложенной в Лекции 2.

# Листинг кода:

#include <iostream>  
#include "fstream"  
#include "vector"  
const int arraySize = 30;  
int steps = 0;  
int LinearSearch(int data[], int value)  
{  
 for (int i = 0; i < arraySize; ++i) {  
 steps++;  
 if(data[i] == value) return i;  
 }  
}  
int Binary(int data[], int key, int l,int r)  
{  
 steps++;  
 int mid = (r+l)/2;  
 if(data[mid] == key) return mid;  
 else if(data[mid] > key) {  
 return Binary(data,key,l,mid);  
 }  
 else{  
 return Binary(data,key,mid+1,r);  
 }  
}  
int BinarySearch(int data[],int value)  
{  
 return Binary  
 (data,value,0, arraySize);  
}  
void BubbleSort(int data[])  
{  
 for (int i = 0; i < arraySize; ++i) {  
 for (int j = i; j < arraySize; ++j) {  
 if(data[i] > data[j])  
 {  
 int tmp = data[i];  
 data[i] = data[j];  
 data[j] = tmp;  
 }  
 }  
 }  
}  
int main() {  
 std::ifstream stream;  
 stream.open("a.txt",std::ios\_base::in);  
 if(!stream)  
 {  
 std::cout << "Can`t open file. Check that file is exists!";  
 return -1;  
 }  
 int data[30];  
 for (int i = 0; i < arraySize; ++i) {  
 stream >> data[i];  
 }  
 stream.close();  
 //Выводим изначальный массив  
 std::cout << "Array from file a.txt = ";  
 for (int i = 0; i <arraySize; ++i) {  
 std::cout << data[i] << " ";  
 }  
 std::cout << std::endl;  
  
 std::cout << "Sorting array...";  
 BubbleSort(data);  
 std::cout << "Sorted array = ";  
 for (int i = 0; i < arraySize; ++i) {  
 std::cout << data[i] << " ";  
 }  
  
  
 std::cout << std::endl << "Let`s find 999 value:" << std::endl << "Linear returns index:" <<LinearSearch(data,999) << std::endl;  
 std::cout << "Steps for linear:" << steps << std::endl;  
 steps =0;  
 std::cout << "Binary returns index:" << BinarySearch(data,999) << std::endl;  
 std::cout << "Steps for binary:" << steps << std::endl;  
 return 0;  
}

# Результат работы программы:



# Вывод:

В ходе выполнения практической работы была написана программа, ищущая число среди 30 других чисел. Бинарный поиск оказался в разы эффективнее. Сложность линейного поиска O(n)

Сложность бинарного поиска O(log(n))