

**МИНОБРНАУКИ РОССИИ**  
**САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ**  
**ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ**  
**«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)**  
**Кафедра МО ЭВМ**

**ОТЧЕТ**  
**по лабораторной работе №1**  
**по дисциплине «Программирование»**  
**Обзор стандартной библиотеки**

Студент гр. 1304

Новицкий М.Д.

Преподаватель

Чайка К.В.

Санкт-Петербург

2018

## **Цель работы.**

Научиться работать со стандартной библиотекой

## **Задание.**

Напишите программу, на вход которой подается массив целых чисел длины 1000, при этом число 0 либо встречается один раз, либо не встречается.

Программа должна совершать следующие действия:

отсортировать массив, используя алгоритм быстрой сортировки (см. функции стандартной библиотеки)

определить, присутствует ли в массиве число 0, используя алгоритм двоичного поиска (для реализации алгоритма двоичного поиска используйте функцию стандартной библиотеки)

посчитать время, за которое совершен поиск числа 0, используя при этом функцию стандартной библиотеки

вывести строку "exists", если ноль в массиве есть и "doesn't exist" в противном случае

вывести время, за которое был совершен двоичный поиск

определить, присутствует ли в массиве число 0, используя перебор всех чисел массива

посчитать время, за которое совершен поиск числа 0 перебором, используя при этом функцию стандартной библиотеки

вывести строку "exists", если 0 в массиве есть и "doesn't exist" в противном случае

вывести время, за которое была совершен поиск перебором.

Результат двоичного поиска, время двоичного поиска, результат поиска перебором и время поиска перебором должны быть выведены именно в таком порядке и разделены символом перевода строки.

## Выполнение работы.

Для выполнения поставленной задачи были использованы заголовочные файлы стандартной библиотеки C: time.h, stdio.h, stdlib.h. С помощью for и scanf идёт считывание ввода с клавиатуры. Затем с помощью qsort идёт сортировка по возрастанию чисел в массиве с помощью функции cmp. Cmp получает на вход две переменные типа void\*. Внутри функции происходит сравнение чисел, если предыдущее число больше следующего, то они меняются местами. Затем вводится переменная a, типа int, ей присваивается значение 0. Чтобы посчитать время выполнения работы создаётся переменная типа clock\_t, time\_1. Производится бинарный поиск 0 в строке с помощью bsearch, значение записывается в переменную index. Если 0 есть в строке, то программа выводит “exists”, если нет, то “doesn’t exist”. Далее выводится время работы. Затем вводится переменная f типа int равная 0, так же заводятся переменные типа clock\_t, time\_2. Далее с помощью for массив проверяется на наличие 0 в нём, если программа его там находит, то цикл завершается и выводит “exists”, в ином случае “doesn’t exist”, после чего выводится время работы программы.

## Тестирование.

Результаты тестирования представлены в табл. 1.

Входные данные	Результат	Комментарий
0 0 1	Exist 0.00002 Exist 0.00001	Корректно
1 2 3	Doesn't exist 0.00002 Doesn't exist 0.00001	Корректно

## Выводы.

Была написана программа, которая использует заголовочные файлы стандартной библиотеки C.

## **ПРИЛОЖЕНИЕ А**

### **ИСХОДНЫЙ КОД ПРОГРАММЫ**

Название файла: lb1.c

```
#include <stdio.h>
#include <stdlib.h>
#include <time.h>

int cmp(const void* a, const void* b){
    const int* x = (const int*)a;
    const int* y = (const int*)b;
    if(*x > *y){
        return 1;
    }
    if(*x < *y){
        return -1;
    }

    return 0;
}

int main(){
    int N = 1000;
    int arr[N];
    for(int i = 0; i < N; i++){
        scanf("%d", &arr[i]);
    }
}
```

```

    }
    qsort(arr,N,sizeof(int),cmp);
    int a = 0;
    float s = clock();
    int* index = bsearch(&a, arr, N, sizeof(int), cmp);
    float e = clock();
    if(index){
        printf("exists\n");
    }
    else{
        printf("doesn't exist\n");
    }
    printf("%f\n", e - s);
    int f = 0;
    float new_s = clock();
    for(int i = 0; i < N; i++){
        if(a == arr[i]){
            f = 1;
            break;
        }
    }
    float new_e = clock();
    if(f){
        printf("exists\n");
    }
    else{
        printf("doesn't exist\n");
    }
    printf("%f\n", new_e - new_s);
    return 0;
}

```