

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

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

Студент гр. 1304

Макки К.Ю

Преподаватель

Чайка К.В.

Санкт-Петербург

2022

Цель работы.

Научиться использовать основные функции стандартной библиотеки Си.

Задание.

Вариант 2 Напишите программу, на вход которой подается массив целых чисел длинны 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.

Подключение заголовочных файлов time.h, stdio.h, stdlib.h. В функции main() используем цикл for с помощью которого считываем массив. С помощью функции qsort массив сортируется по возрастанию, используя функцию strcmp. Внутри функции strcmp происходит сравнение двух чисел, поданных на вход и возвращается одно из трех значений: -1 – если первое число меньше второго, 0 – если они равны, 1 – если первое число больше

второго. Вводится переменная `key` в котором значение 0. Чтобы узнать время работы программы используется переменные в которых храниться значение функции `clock`. Потом используем бинарный поиск через `bsearch` с помощью стр ищем число 0. Получаем время от разности между текущим значением функции `clock` и предыдущим значением этой переменной. Значение, которое возвращает функция `bsearch` записывается в переменную `index` типа `int*`. с помощью конструкции `if-else`, которой на вход подается `index`, выводится соответствующая строка `exists` – если значение `index` не равно `NULL`, `doesn't exist` – в ином случае.

Тестирование

Таблица Б.1 - Примеры тестовых случаев

№ п/п	Входные данные	Выходные данные	Комментарии
1.	77 48 65 79 66 31 60 36 25 33	doesn't exist 7 doesn't exist 4	n=10
2.	98 60 86 91 88 53 82 57 47 45 08 76 09 48 34 35 69 33 57 61 48 26 52 90 0 18 78 47 69 01 94 75 29 93 90 24 81 86 15 51 70 88 38 19 32 71 64 36 92 64 50 95 51 56 86 23 43 49 01 70 03 34 27 09 62 36 23 55 24 99 46 84 74 82 72 43 63 74 67 79 25 67 93 28 84 69 19 54 44 52 35 83 36 44 38 39 95 37 53 68	exists 6 exists 4	n=100

Выводы.

В ходе выполнения лабораторной работы были изучены возможности, которые предоставляет стандартная библиотека языка Си. Была написана программа, сортирующая массив из 1000 целых чисел по возрастанию и выясняющая есть ли среди данных чисел 0 с помощью двух вариантов

поиска: используя bsearch и перебором, - а также выводящая время, за которое были выполнены обе операции.

ПРИЛОЖЕНИЕ А

ИСХОДНЫЙ КОД ПРОГРАММЫ

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

int cmp(const void* a, const void* b){
    const int* f = (const int*)a;
    const int* s = (const int*)b;
    if(*f < *s)
        return -1;
    if(*f > *s)
        return 1;
    return 0;
}

int main(){
    int n = 1000;
    int key = 0;
    int cond = 0;
    int arr[n];

    for(int i = 0; i < n; i++)
        scanf("%d", &arr[i]);

    qsort(arr, n, sizeof(int), cmp);

    clock_t start = clock();
    int* tmp = bsearch(&key, arr, n, sizeof(int), cmp);
    clock_t end = clock();

    if(tmp)
        printf("exists\n");
    else
        printf("doesn't exist\n");

    printf("%ld\n", end - start);

    clock_t start_2 = clock();

    for(int i = 0; i < n; i++){
        if(key == arr[i]){
            cond = 1;
            break;
        }
    }
}
```

```
clock_t end_2 = clock();  
  
if(cond)  
    printf("exists\n");  
else  
    printf("doesn't exist\n");  
  
printf("%ld\n", end_2 - start_2);  
return 0;  
}
```