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

Вариант 2

Студент гр. 0382 Афанасьев Н. С.

Преподаватель Берленко Т. А.

Санкт-Петербург
2021

Цель работы.

Изучение возможностей стандартной библиотеки языка C и их использование в решении задачи.

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.

Выполнение работы.

Функция *int* **findBin***(int* arr)* выполняет бинарный поиск числа 0 в массиве, используя функцию *bsearch* стандартной библиотеки.

Функция *int* **findBrute***(int* arr)* выполняет поиск числа 0 в массиве методом перебора, через цикл *for*. Обе функции принимают массив чисел, возвращают 1, если ноль найден, иначе 0.

Функция *int* **compare***(const void* a, const void* b)* выполняет сравнение двух чисел, возвращая разницу двух чисел. Это функция необходима в качестве аргумента для функций *bsearch* и *qsort*.

Функция *void* **stopwatch***(int* arr, int (*f)(int*))* принимает в качестве аргументов массив чисел, и другую функцию *f* (предполагается *findBrute* или *findBin*). Для начала, считывается текущее количество тактов процессора с помощью функции *clock* библиотеки *time.h* и записывается в переменную time типа *clock_t*. Далее выполняется и выводится результат функции *f* (“*exists*”/”*doesn’t exist*”). В конце выводится время выполнения функции в секундах, которое высчитывается по формуле *=((float)(clock()-time))/CLOCKS_PER_SEC*.

В функции *int* **main***()* считывается массив *arr* целых чисел размера *SIZE*(=1000). Далее этот массив сортируется с помощью функции *qsort* стандартной библиотеки, выполняется функция *stopwatch*, сначала для *findBin*, а затем для *findBrute*. Если программа завершилась успехом, возвращается 0.

Разработанный программный код см. в приложении А.

Тестирование.

Результаты тестирования представлены в табл. 1.

Таблица 1 – Результаты тестирования

№ п/п	Входные данные	Выходные данные	Комментарии
1.	Все чётные числа от -1000 до 998	exists 0.000000 exists 0.000000	Верно Время колеблется от 0 до 0.01с для обоих методов (CLOCKS_PER_SEC = 1000)
2.	Все нечётные числа от -999 до 999	doesn't exist 0.000000 doesn't exist 0.000000	Верно

Выводы.

Был изучена работа со стандартной библиотекой языка С.

Разработана программа, которая считывает массив из 1000 целых чисел, сортирует список по возрастанию, используя алгоритм быстрой сортировки, находит 0 в массиве, если он есть, двумя способами: бинарным поиском и методом перебора – в обоих случаях высчитывается время выполнения поиска. Программа выводит результат вида: результат двоичного поиска, время двоичного поиска, результат поиска перебором и время поиска перебором.

ПРИЛОЖЕНИЕ А

ИСХОДНЫЙ КОД ПРОГРАММЫ

Название файла: main.c

```
#include <stdio.h>
#include <stdlib.h>
#include <time.h>
#define SIZE 1000

int compare(const void* a, const void* b){
    return *(int*)a - *(int*)b;
}

int findBin(int* arr){
    const int zero = 0;
    return (bsearch(&zero, arr, SIZE, sizeof(int), compare) != NULL);
}

int findBrute(int* arr){
    for(short i = 0; i < SIZE; i++) if(arr[i] == 0) return 1;
    return 0;
}

void stopwatch(int* arr, int (*f)(int*)){
    clock_t time = clock();
    printf(f(arr) ? "exists\n" : "doesn't exist\n");
    printf("%f\n", ((float)(clock()-time))/CLOCKS_PER_SEC);
}

int main(){
    int arr[SIZE];
    for(short i = 0; i < SIZE; i++) scanf("%d", &arr[i]);
    qsort(arr, SIZE, sizeof(int), compare);
    stopwatch(arr, findBin);
    stopwatch(arr, findBrute);
    return 0;
}
```