

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

ОТЧЕТ
по лабораторной работе №4
по дисциплине «Программирование»
Тема: Обзор стандартной библиотеки

Студент гр. 9304

Мохаммед А.А.

Преподаватель

Чайка К.В.

Санкт-Петербург

2019

Цель работы.

Познакомиться с некоторыми функциями стандартной библиотеки.

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.

Основные теоретические положения.

Стандартной библиотекой языка Си (также известная как `libc`, `crt`) называется часть стандарта ANSI C, посвященная заголовочным файлам и библиотечным подпрограммам. Является описанием реализации общих операций, таких как обработка ввода-вывода и строк, в языке программирования Си. Стандартная библиотека языка Си — это описание программного интерфейса, а не настоящая библиотека, пригодная для использования в процессе компиляции.

Стандартная библиотека ANSI Си состоит из 24 заголовочных файлов, каждый из которых можно подключать к программному проекту при помощи одной директивы. Каждый заголовочный файл содержит объявления одной или более функций, определения типов данных и макросы.

Существует не одна реализация стандартной библиотеки: так, например, стандартная библиотека для систем Linux (GNU Libc) включает функции, которые могут быть недоступны в других реализациях.

Выполнение работы.

Разработанный алгоритм решает поставленную задачу, используя функции стандартной библиотеки. Алгоритм составлен следующим образом:

- 1) Программа считывает элементы статического массива длины 1000;
- 2) Сортирует элементы введенного массива, используя библиотечную функцию `qsort`. Для сравнения двух элементов массива написана функция-компаратор.
- 3) Сохраняет в переменную `start` время, прошедшее с запуска программы, с помощью функции `clock` стандартной библиотеки;

4) Производит бинарный поиск нуля в массиве, сохраняя его адрес в переменную ptr, после чего выводит сообщение «exists», если ноль найден, и «doesn't exist», если ноль не найден;

5) Сохраняет в переменную end время с запуска программы и выводит разность времён end и start, делённую на макрос CLOCKS_PER_SEC (количество машинных тактов в секунду для данной машины), т. е. Время в секундах;

6) Ищет ноль в массиве перебором всех его элементов, при этом аналогично замеряя время, потребовавшееся для этого, и выводит сообщение «exists» или «doesn't exist» в зависимости от результата поиска, а также затраченное время.

Тестирование.

Результаты тестирования программы (для длины массива $N = 10$) представлены в табл. 1.

Таблица 1 – Результаты тестирования

№ п/п	Входные данные	Выходные данные	Комментарии
1.	-10 216 -17 0 2 33 7 80 -9 1	exists 0.000000 exists 0.000000	Получен ожидаемый результат. Здесь и далее время на перебор 10 элементов оказалось слишком незначительным, чтобы разность end и start была отлична от 0. На моём компьютере время, не равное 0, выводится при $N > 10000$; в проверяющей системе stepik — уже при $N = 1000$
2.	-100 -15 10 -20 -5 -70 -8 -78 -8 -1	doesn't exist 0.000000	Получен ожидаемый результат.

		doesn't exist 0.000000	
3.	1 2 3 4 5 6 7 8 9 0	exists 0.000000 exists 0.000000	Получен ожидаемый результат.

Выводы.

В ходе лабораторной работы была написана программа, выполняющая поиск определённого элемента (нуля) в массиве целых чисел и замеряющая ушедшее на это время с использованием функций стандартной библиотеки.

ПРИЛОЖЕНИЕ А

ИСХОДНЫЙ КОД ПРОГРАММЫ

Название файла: main.c

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

void fill_arr(int* ptr, int l){
    for (int i = 0; i < l; i++){
        scanf("%d", ptr + i); }}

int int_cmp(const void* x, const void* y){
    x = *((int*)x);
    y = *((int*)y);
    if(x == y) return 0;
    if(x < y) return -1;
    if(x > y) return 1;}

int main(){
    int arr[N];
    fill_arr(arr, N);
    int zero = 0;

    qsort(arr, N, sizeof(int), int_cmp);
    clock_t start = clock();
    int* ptr = (int*)bsearch(&zero, arr, N, sizeof(int), int_cmp);

    clock_t end = clock();

    if (ptr) puts("exists");
    else puts("doesn't exist");

    printf("%lf\n", (end - start)/(double)CLOCKS_PER_SEC);

    start = clock();

    for (int i = 0; i < N; i++){
        if (arr[i] == 0){
            puts("exists");
            break;}
        if(i == N-1)puts("doesn't exist");}

    end = clock();
    printf("%lf\n", (end - start)/(double)CLOCKS_PER_SEC);

    return 0;
}
```