

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.

Студентка гр. 1304

Чернякова В.А.

Преподаватель

Чайка К.В.

Санкт-Петербург

2022

Цель работы.

Освоение работы со стандартными библиотеками языка C на примере использующей их программы. Научиться пользоваться функциями данных библиотек для написания кода программы.

Задание.

Вариант 2.

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 корректной работы программы были подключены стандартные библиотеки языка `<stdio.h>`, `<stdlib.h>` и `<time.h>`.

В главной функции `int main ()` объявляются переменные целочисленного типа данных `int`: массив `arr[1000]`, в котором будут храниться значения, вводимые с клавиатуры, массива длиной 1000, переменная `cnt_zero = 0`, которая отвечает за количество «0» в массиве, и переменная `key = 0`, в которой хранится значение, нахождение или отсутствие которого в массиве нужно определить.

Благодаря подключенной стандартной библиотеке языка C `<time.h>` объявляются переменные типа `clock_t`, в которых будет храниться значение затраченного времени на поиск искомого значения в массиве с помощью бинарного поиска и методом перебора - `t1` и `t2` соответственно.

С помощью цикла `for` и функции `scanf` происходит считывание 1000 элементов массива.

С помощью функций стандартной библиотеки применяется функция быстрой сортировки `qsort(arr, 1000, sizeof(int), cmp)`. Где первое значение `arr` – указатель на массив, в котором необходимо отсортировать элементы, второе `1000` – количество элементов массива, далее `sizeof(int)` – размер его элементов, последнее `cmp` – функции сравнения, позволяющая отсортировать массив должным образом.

Переменной `t1` присваивается значение работы функции стандартной библиотеки `clock()`, которая возвращает значение, которое приблизительно соответствует времени работы вызывающей программы, а именно времени работы двоичного поиска.

Так как результат двоичного поиска – указатель, создается переменная `result` типа `int *` и присваивается значение `(int*) bsearch(&key, arr, 1000, sizeof(int), cmp)`. В функции двоичного поиска: первое значение `&key` – адрес искомого элемента, второе `arr` – указатель на массив, третье `1000` – размер

массива, далее *sizeof(int)* – размер элементов массива, последнее *cmp* – функция сравнения.

Для определения точного времени, потраченного на поиск искомого значения с помощью функции двоичного поиска переменной *t1* присваивается значение равное *clock()* - *t1*.

Далее при нахождении искомого значения массива с помощью функции *printf* на экран выводится "*exists*", иначе "*doesn't exist*". Также выводится значение равное времени работы поиска *printf("%ld",t1)*.

Переменной *t2* присваивается значение работы функции стандартной библиотеки *clock()*, которая возвращает значение, которое приблизительно соответствует времени работы вызывающей программы, а именно времени работы двоичного поиска.

С помощью цикла *for* перебираются все числа массива для поиска «0»: если условие *arr[i] == 0* выполняется, то переменная, отвечающая за количество «0» в массиве, *cnt_zero* увеличивается.

Для определения точного времени, потраченного на поиск искомого значения с помощью функции двоичного поиска переменной *t2* присваивается значение равное *clock()* – *t2*.

При выполнении условия (*cnt_zero != 0*) на экран с помощью функции *printf* выводится "*exists*", иначе "*doesn't exist*". Также выводится значение равное времени работы поиска *printf("%ld",t2)*.

Функция *cmp*:

Функция компаратор реализуется для работы стандартных функций библиотеки языка по необходимому принципу. Функция *int cmp (const void *a, const void *b)* принимает на вход указатели на переменные типа *const void*. Для быстрой сортировки первый элемент и второй – идущие друг за другом значения в массиве. Для бинарного поиска первый элемент, принимаемый функцией – ключ, то есть искомое значение, второй – очередной элемент массива.

В переменную `const int *f` записывается значение, которое хранится в адресе указателя `(const int *) a`. Аналогично `const int *s = (const int *) b`.

Далее, используя оператор разыменования `*`, происходит сравнение элементов и возвращение `1`, `-1` или `0` в зависимости от выполняемого условия.

Благодаря такому сравнению каждого элемента `qsort` – сортирует массив по возрастанию, `bsearch` – ищет необходимое значение.

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.	12 3 0 87 2 13 2 2 0 19	exists 1 exists 0	Программа работает корректно.
2.	12 734 182 364 19 2 3 4 5 12 3 4 123 4 12 87 19 23 57 8	doesn't exist 2 doesn't exist 1	Программа работает корректно.

Выводы.

В ходе лабораторной работы были освоены основы работы с функциями стандартных библиотек языка программирования С. Написана программа, производящая поиск в массиве значения «0» двумя способами: с помощью функции бинарного поиска стандартной библиотеки и методом перебора – и сравнивающая значение времени работы каждого из способов.

ПРИЛОЖЕНИЕ А

ИСХОДНЫЙ КОД ПРОГРАММЫ

Название файла: Chernyakova_Valeria_lb1/main.c

```
#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 arr[1000], cnt_zero = 0, key = 0;
    clock_t t1, t2;
    for (int i = 0; i < 1000; i++)
        scanf ("%d", &arr[i]);
    qsort(arr, 1000, sizeof(int), cmp);
    t1 = clock();
    int *result = (int*) bsearch(&key, arr, 1000, sizeof(int),
cmp);

    t1 = clock() - t1;
    if (result)
        printf("exists\n");
    else
        printf ("doesn't exist\n");
    printf ("%ld\n",t1);
    t2 = clock();
    for (int i = 0; i < 1000; i++)
        if (arr[i] == 0)
            cnt_zero++;
    t2 = clock()-t2;
    if (cnt_zero != 0)
        printf("exists\n");
    else
        printf ("doesn't exist\n");
    printf ("%ld",t2);
    return 0;
}
```