**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: Стандартная библеотека Си**

|  |  |  |
| --- | --- | --- |
| Студент гр. 6303 |  | Архипцев Евгений |
| Преподаватель |  | Чайка Константин |

Санкт-Петербург

2017

**Цель работы:**

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.

**Ход Работы:**

**1. Функция Bsearch**

Функция bsearch выполняет двоичный поиск в массиве.

Указатель на элемент массива, который равен значению в key.  
Если искомый элемент  не найден, возвращается нулевой указатель.

**2.Функция Qsort**

Функция qsort выполняет сортировку элементов массива, на который ссылается указатель first. Для каждого элемента массива устанавливается размер в байтах, который передается через параметр size. Последний параметр функции qsort — указатель comparator на функцию сравнения, которая используется для определения порядка следования элементов в отсортированном массиве.

**3.Функция Clock()**

Возвращает количество временных тактов, прошедших с начала запуска программы. С помощью [макроса CLOCKS\_PER\_SEC](http://cppstudio.com/spravochnik/standartnye-zagolovochnye-fajly-iz-si-v-s/zagolovochnyj-fajl-ctime-time-h/makros-clocks_per_sec/)функция получает количество пройденных тактов за 1 секунду. Таким образом, зная сколько выполняется тактов в секунду, зная время запуска программы можно посчитать время работы всей программы или отдельного её фрагмента, что и делает данная функция.

**Исходный код программы:**

#include <stdio.h>

#include <stdlib.h>

#include <time.h>

#define For(i,a,b) for (i=a;i<b;i++)

const int N = 1000;

int compare(const void \* x1, const void \* x2) // функция сравнения элементов массива

{

return ( \*(int\*)x1 - \*(int\*)x2 ); // если результат вычитания равен 0, то числа равны, < 0: x1 < x2; > 0: x1 > x2

}

int main() {

int \*a ,end,start,i; // e,s переменные для подсчета времени

a = (int\* )malloc(N\*sizeof(int));

For(i,0,N){ //считываем массив

scanf("%d",&a[i]);

}

/\*quicksort\*/

qsort(a, N, sizeof(int), compare);

/\*\*/

/\*find 0 Binary search|| times\*/

start = clock();

int number = 0;

int \* ptrItem = (int\*) bsearch(&number, a, N, sizeof (int), compare);

end = clock();

/\*\*/

/\*\*/

if (ptrItem != NULL){

printf("exists\n");}

else{

printf("doesn't exist\n");}

printf("%d",end-start);

/\*\*/

/\*find 0 Full search\*/

//bool Fd\_Nm = false;

int Fd\_Nm = 0;

start = clock();

For(i,0,N){

if (a[i] == 0){

/\*Fd\_Nm = true;\*/Fd\_Nm = 1;break;

}

}

end = clock();

if (Fd\_Nm == 1){

printf("exists\n");

}else{

printf("doesn't exist\n");

}

printf("%d",end-start);

/\*\*/

free(a);

return 0;

}

**Вывод:** Были получены знания в области использования функций Bsearch и Qsort. Изучена функция Clock() макрос CLOCKS\_PER\_SEC.