**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 |  | Погодаев Н.М. |
| Преподаватель |  | Берленко Т.А. |

Санкт-Петербург

2017

**Цель работы:**

Написать 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.

**Описание работы программы.**

Программа представлена в виде нескольких функций:

1. Вспомогательных, таких как int comp (int\* a, int\* b) и int funccmp (void\* x1, void\* x2)
2. Основная int main()которая заполняет массив, выполняет сортировку массива, поиск элемента в массиве и замеряет время работы функций поиска элемента

**Исходный код программы:**

#include <stdio.h>

#include <stdlib.h>

#include <time.h>

#define N 1000

int funccmp( const void \*x1, const void \*x2){ //функция сравнения двух эл-тов

return (\*(int\*)x1 - \*(int\*)x2);

}

int comp (const int \*a, const int \*b)//компаратор для qsort

{

return \*a - \*b;

}

int main() {

int i;

int timeBeg1, timeBeg2, timeEnd1, timeEnd2;

int key=0;//ключ для двоичного поиска

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

for (i=0;i<N;i++)//заполнение массива числами

{

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

}

qsort(arr, N, sizeof(int),(int(\*)(const void \*,const void \*))comp); //сортировка массива через алгоритм быстрой сортировки

timeBeg1 = clock(); //время начала поиска

int \*ptrItem=(int\*) bsearch( &key, arr, 100, sizeof(int), funccmp);

timeEnd1 = clock(); //время конца поиска

if (ptrItem==NULL) { //вывод результата двоичного поиска

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

else {

printf("exists\n");}

printf("%d\n",timeEnd1-timeBeg1);//вывод времени двоичного поиска

int l = 0; //поиск 0 перебором

timeBeg2 = clock();

for (i=0;i<N;i++)

{

if (arr[i]==key)

{

l=1;

}

}

timeEnd2 = clock();

if (l==1)//вывод результата поиска перебором

{

printf("exists\n");

}

else

{

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

}

printf("%d\n",timeEnd2-timeBeg2);//вывод времени поиска перебором

free(arr);

return 0;

}