**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**

|  |  |  |
| --- | --- | --- |
| Студент гр. 1384 |  | Усачева Д.В. |
| Преподаватель |  | Жангиров Т. Р. |

Санкт-Петербург 2022

**Цель работы.**

Целью работы является ознакомление с наиболее часто используемыми и полезными функциями стандартной библиотеки языка C, а также работа со временем.

**Задание.**

(Вариант 3)

Напишите программу, на вход которой подается массив целых чисел длины **1000**.

Программа должна совершать следующие действия:

* отсортировать массив с помощью алгоритма "сортировка пузырьком"
* посчитать время, за которое будет совершена сортировка, используя при этом **функцию стандартной библиотеки**
* отсортировать массив с помощью алгоритма "быстрая сортировка" (quick sort), используя при этом **функцию стандартной библиотеки**
* посчитать время, за которое будет совершена сортировка, используя при этом **функцию стандартной библиотеки**
* вывести отсортированный массив (элементы массива должны быть разделены пробелом)
* вывести время, за которое была совершена сортировка пузырьком
* вывести время, за которое была совершена быстрая сортировка

*Отсортированный массив, время сортировки пузырьком, время быстрой сортировки должны быть выведены с новой строки, при этом элементы массива должны быть разделены пробелами.*

**Выполнение работы.**

Основное тело программы, функция main, начинается с заполнения двух массивов bub\_sorted и q\_sorted с консоли. После чего измеряется прошедшее время с запуска программы и затем выполняется первая сортировка с одним массивом, после чего вновь засекается время и выполняется вторая сортировка для другого массива. После вновь засекается время. После выводят один из отсортированных массивов, а после них, путём вычитания известным нам значений время и делениям для перевода из тиков в секунды, показывается время сортировки bub\_sort и qsort.

Массив bub\_sorted сортируется при помощи «сортировки пузырьком». Сортировка пузырьком работает следующим образом: программа проходится по массиву, сравнивая два соседних элемента. При необходимости она меняет их местами. Дойдя до конца массива программа возвращается в начало массива и повторяет алгоритм, пока массив не будет полностью отсортирован.

Массив q\_sorted сортируется быстрой сортировкой при помощи встроенной функции qsort.

Для измерения времени каждого из алгоритмов используется функция clock.

Разработанный программный код см. в приложении 1.

**Выводы.**

Были изучены наиболее часто используемые и полезные функциями стандартной библиотеки языка C, а также работа со временем.

**ПРИЛОЖЕНИЕ 1**

**ИСХОДНЫЙ КОД ПРОГРАММЫ**

Название файла: main.c

#include <stdlib.h>

#include <stdio.h>

#include <stdbool.h>

#include <time.h>

void bub\_sort(int \*arr){

bool flag = false;

while(!flag){

flag = true;

for(int i=0; i < 1000-1; i++){

if (arr[i]>arr[i+1]){

flag = false;

int a = arr[i];

arr[i] = arr[i+1];

arr[i+1] = a;

}

}

}

}

int cmp(const void\*a, const void\*b){

return \*(int\*)a - \*(int\*)b;

}

int main()

{

int q\_sorted[1000];

int bub\_sorted[1000];

for(int i=0; i<1000; i++){

int a;

scanf("%d ", &a);

q\_sorted[i] = a;

bub\_sorted[i] = a;

}

clock\_t q\_start = clock();

qsort(q\_sorted, 1000, sizeof(int), cmp);

clock\_t q\_end = clock();

float q\_time = (float)(q\_end - q\_start)/CLOCKS\_PER\_SEC;

clock\_t b\_start = clock();

bub\_sort(bub\_sorted);

clock\_t b\_end = clock();

float bub\_time = (float)(b\_end - b\_start)/CLOCKS\_PER\_SEC;

for(int i=0; i<1000; i++){

printf("%d ", bub\_sorted[i]);

}

printf("\n%f", bub\_time);

printf("\n%f ", q\_time);

return 0;

}