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

ОТЧЕТ

по лабораторной работе №3
по дисциплине «Программирование»
Тема: Структуры и обзор stdlib.

Студент гр. 1304 _____ Стародубов М.В.

Преподаватель _____ Чайка К.В.

Санкт-Петербург
2022

Цель работы.

Узнать, какие возможности предоставляет стандартная библиотека языка программирования Си, научиться использовать основные функции стандартной библиотеки.

Задание.

Вариант – 4.

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

Программа должна совершать следующие действия:

- отсортировать массив по невозрастанию модулей элементов 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 быстрая сортировка

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

Выполнение работы.

Для корректрой работы программы в файл исходного кода необходимо подключить файлы стандартной библиотеки stdio.h, stdlib.h, time.h. Также в заголовке файла исходного кода определяется значение SIZE, равное 1000.

Выполнение программы начинается с объявления статического массива размера SIZE, данный массив заполняется числами, которые подаются на вход программе.

Для корректной работы быстрой соктировки необходимо реализовать функцию-компоратор. Выполение функции-компоратора начинается с того, что в переменной first записывается модуль числа, которое храниться по адресу, сохраненному в первом аргументе функции, а в переменную second записывается модуль числа, которое храниться по адресу, сохраненному во втором аргументе функции. Далее производиться сравнение данных значений: если значение, сохраненное в first больше, чем значение, сохраненное в second, то функция возвращает -1, иначе если значение, сохраненное в first меньше, чем значение, сохраненное в second, то функция возвращает 1, иначе функция возвращает 0.

В переменной before с помощью функции clock сохраняется количество тактов прояессора, совершенных с начала работы программы. После выполнения быстрой сортировки в переменную after также сохраняем количество тактов прояцессора, т.о. разность значений переменных before и after – это количество тактов процессора, затраченное на совершение быстрой сортировки.

Далее программа выводит отсортированный массив и время, затраченное на его сортировку. Чтобы корректно вывести время, затраченное программой на сортировку, необходимо привести разность значений переменных before и after к типу float и разделить на константу CLOCKS_PER_SEC – количество тактов прочессора, совершаемых за одну секунду.

Выводы.

В ходе выполнения лабораторной работы были изучены возможности, предоставляемые стандартной библиотекой языка Си. Был создан программный продукт, производящий сортировку идущего на вход программе массива целых чисел по невозрастанию модулей элементов массива, программа возвращает отсортированный массив и время, за которое была выполнена сортировка.

ПРИЛОЖЕНИЕ А

ИСХОДНЫЙ КОД ПРОГРАММЫ

Название файла: main.c

```
#include <stdio.h>
#include <stdlib.h>
#include <time.h>

#define SIZE 1000

int cmp(const void *a, const void *b)
{
    int first = abs(*((int *) a));
    int second = abs(*((int *) b));

    if (first > second)
    {
        return -1;
    } else if (second > first)
    {
        return 1;
    }
    return 0;
}

int main()
{
    int arr[SIZE];
    for (int i = 0; i < SIZE; i++)
    {
        scanf("%d", &arr[i]);
    }

    clock_t before = clock();
    qsort(arr, SIZE, sizeof(int), cmp);
    clock_t after = clock();

    for (int i = 0; i < SIZE; i++)
    {
        printf("%d ", arr[i]);
    }
    printf("\n%f", (after - before)/CLOCKS_PER_SEC);


    return 0;
}
```