

**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. 1304

\_\_\_\_\_

Басыров В.А.

Преподаватель

\_\_\_\_\_

Чайка К.В.

Санкт-Петербург

2022

### **Цель работы.**

Изучение возможностей языка, а также углубленное изучение библиотеки `time.h`, практика работы со временем, и изучение функции `qsort`.

### **Задание.**

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.

### **Основные теоретические положения.**

`void qsort ( void * first, size_t number, size_t size, int ( * comparator ) ( const void *, const void * ) );` - функция сортировки, где `first` — указатель на 1 элемент массива, `number` — количество элементов, `size` — размер одного элемента, а оставшийся элемент — функция компаратор( функция сравнения).

`void time(NULL)` — показывает текущее время.

`Void difftime(time2, time1)` — вычисление разности времени.

### **Выполнение работы.**

Считывается массив, замеряется текущее время, далее производится сортировка, еще раз считывается текущее время, а дальше вычитывается из 1 времени второе.

### **Тестирование.**

Здесь результаты тестирования, которые помещаются на одну страницу.

Результаты тестирования представлены в табл. 1.

Таблица 1 – Результаты тестирования

№ п/п	Входные данные	Выходные данные	Комментарии
1.	Числа [1,1000]	0	
2.	Числа [256,1256]	0	
3.	Числа [1000,1] в обр порядке	0	
4	Числа [1256,256] в обр порядке	0	

### **Выводы.**

Были изучены основные библиотеки Си, в частности работа со временем, проанализировано время работы быстрой сортировки.

## ПРИЛОЖЕНИЕ А

### ИСХОДНЫЙ КОД ПРОГРАММЫ

Сначала указываем имя файла, в котором код лежит в репозитории:

Название файла: main.c

```
#include <stdio.h>
#include <stdlib.h>
#include <time.h>
#define N 1000
int cmp(const void *first,const void *second)
{   int a=((int*) first);
    int b=((int*) second);
    if (a<b)
        return 1;
    else
        if (a>b)
            return -1;
        else
            return 0;
}

int main() {
    int *a=malloc(sizeof(int)*N);
    for (int i=0;i<N;i++)
        scanf("%d",&a[i]);
    time_t t1=time(NULL),t2;
    qsort(a,N,sizeof(int),cmp);
    t2=time(NULL);
    for (int i=0;i<N;i++)
        printf("%d ",a[i]);
    printf("\n");
    printf("%f\n",difftime(t2,t1));
}
```