

**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 языка Си.**

Студент гр. 0382

\_\_\_\_\_

Санников В.А.

Преподаватель

\_\_\_\_\_

Берленко Т.А.

Санкт-Петербург

2021

## Цель работы.

Изучение возможностей стандартной библиотеки языка Си.

## Задание.

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.*

## Основные теоретические положения.

**stdio.h:** стандартная библиотека вводы-вывода.

**stdlib.h:**

- Функции для сортировки и поиска.

**qsort:**

```
void qsort (void* base, size_t num, size_t size, int (*compar)(const void*,const void*));
```

Функция принимает указатель на начальный элемент массива, количество элементов и размер одного элемента, а также указатель на функцию для сравнения двух элементов.

Так как тип элементов может быть любым, то и указатель на первый элемент массива имеет тип `void`. Это позволяет, зная адрес первого элемента и размер каждого элемента вычислить адрес любого элемента массива в памяти и обратиться к нему. Остается только сравнить 2 элемента имея 2 указателя на них. Это выполняет функция `compar`, указатель на которую передается функции `qsort` в качестве одного из параметров.

Функция `compar` принимает 2 указателя типа `void`, но в своей реализации может привести их к конкретному типу (так как её реализация остается за программистом, он точно знает элементы какого типа он сортирует) и сравнивает их. Результат сравнения определяется знаков возвращаемого функций `qsort` числа.

**time.h:** заголовочный файл стандартной библиотеки языка программирования C, содержащий типы и функции для работы с датой и временем.

**Выполнение работы.****Ход работы:**

Объявляем нужные нам заголовочные файлы (их 3), объявляем константу `LEN=1000` и переменные:

`int i,j` — для итераций циклов.

`char sep` — разделитель.

`clock_t time` — время, прошедшее с момента запуска программы.

`float time1, time2` — время быстрой сортировки и «методом пузырька» соответственно.

int noSwap — для проверки на условия: предыдущий > следующего.

int st - «стакан», для перемещения предыдущего элемента на место следующего и наоборот.

Далее с помощью цикла while() вводим с клавиатуры массив длины 1000.

После этого происходит быстрая сортировка и считается время, за которое она выполнится. Записываем время в переменную time1 с помощью макроса CLOCKS\_PER\_SEC, который позволяет определить такты системных часов в секунды.

Далее идет «сортировка пузырьком» и считается время, за которое она выполнится. Записываем время в переменную time2 с помощью макроса CLOCKS\_PER\_SEC, который позволяет определить такты системных часов в секунды.

Выводим отсортированный массив и время сортировок с новой строки.

### **Тестирование.**

Результаты тестирования представлены в табл. 1 при значении LEN 8.

Таблица 1 – Результаты тестирования

№ п/п	Входные данные	Выходные данные	Комментарии
1.	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0.000001 0.000002	Программа работает верно
2.	-100 1234 9540 10 0 -340 4392 12	-340 -100 0 10 12 1234 4392 9540 0.000005 0.000001	Программа работает верно
3.	8391238 -1293434 123039 -1239 129034 19320 - 123903 999999	-1293434 -123903 -1239 19320 123039 129034 999999 8391238 0.000008 0.000001	Программа работает верно

### **Выводы.**

Был изучен и освоен функционал стандартной библиотеки языка программирования Си.

Разработана программа, принимающая на вход массив целых чисел, сортирующая его двумя методами (быстрой сортировкой и «методом пузырька»), а так же посчитавшая время быстрой сортировки и «методом пузырька».

## ПРИЛОЖЕНИЕ А

### ИСХОДНЫЙ КОД ПРОГРАММЫ

#### Название файла: main.c

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

#define LEN 1000

int cmp(const void * x1, const void * x2){
    return (*(int*)x1 - *(int*)x2);
}

int main(){
    int i=0, j, mas[LEN], noSwap, st;
    char sep = ' ';
    clock_t time;
    float time1, time2;

    while(i<LEN && sep == ' '){
        scanf("%d%c", &mas[i], &sep);
        i++;
    }

    time = clock();
    qsort(mas, LEN, sizeof(int), cmp); //быстрая
    time = clock() - time;

    time1 = ((float) time)/CLOCKS_PER_SEC;

    time = clock();
    for (i=LEN-1; i >= 0; i--){ //пузырьком
        noSwap=1;
        for (j=0; j < i; j++){
            if (mas[j] > mas[j+1]){
                st = mas[j];
                mas[j] = mas[j+1];
                mas[j+1] = st;
                noSwap=0;
            }
        }
        if (noSwap == 1)
            break;
    }
    time = clock() - time;

    time2 = ((float) time)/CLOCKS_PER_SEC;

    for (i=0; i < LEN; i++){
        printf("%d ", mas[i]);
    }

    printf("\n%f", time1);
```

```
    printf("\n%f", time2);  
    return 0;  
}
```