

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

**ОТЧЕТ**  
**по лабораторной работе №4**  
**по дисциплине "Программирование"**  
**Тема: Обзор стандартной библиотеки**

Студент гр. 9304

Атаманов С.Д.

Преподаватель

Чайка К.В.

Санкт-Петербург

2019

### **Цель работы.**

Написать программу на языке программирования **Си**, используя функции стандартной библиотеки языка Си.

### **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.

Вначале были подключены библиотеки **stdio.h**, **stdlib.h**, **time.h** и объявлен макрос **N**, который заменит все найденные в коде **N** на 1000(максимальный размер исходного массива).

**int main().**

Объявлен целочисленный массив **arr[N]**, который имеет 1000 элементов и целочисленный массив **bubble\_arr[N]**, который также имеет 1000 элементов и будет хранить в себе отсортированный пузырьковым методом исходный массив. Также объявлена целочисленная переменная-маркер **num = 1**, которая участвует в сортировке пузырьковым методом, и целочисленная переменная **exchng**, которая используется для того, чтобы поменять местами два элемента массива в сортировке пузырьковым методом. И две переменные типа **float: bubble\_time** и **qsort\_time**, которые хранят в себе время в секундах, которое потребовалось для сортировки массивов пузырьковым методом и с помощью функции **qsort**.

С помощью цикла **for** и функции **scanf** выполняется заполнение массива **arr[N]** и **bubble\_arr**.

Далее начинается отсчет времени функцией библиотеки **time.h – clock()** и выполняется сортировка пузырьковым методом с помощью цикла **for**. После время переводится в секунды с помощью деления значения, которое вернула функция **clock()** на макрос **CLOCKS\_PER\_SEC** и записывается в переменную **bubble\_time**.

Далее с помощью функции **clock()** начинается отсчет времени сортировки с помощью функции **qsort**. В переменную **qsort\_time** записывается время, за которое была произведена сортировка.

С помощью цикла **for** и функции **printf()** производится вывод на экран отсортированного массива и времени, которое было затрачено на сортировку.

**int cmp().**

Возвращает:

- **-1**, если левый элемент массива **меньше** правого.
- **0**, если элементы **равны**.
- **1**, если левый элемент **больше** правого.

Разработанный программный код см. в Приложении А.

### **Выводы.**

Была написана программа на языке **Си** с использованием функций стандартной библиотеки языка **Си**.

## ПРИЛОЖЕНИЕ А

### ИСХОДНЫЙ КОД ПРОГРАММЫ

Название файла: *main.c*

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

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

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

```
#define N 1000
```

```
int cmp(const void* num1, const void* num2){  
    if(*(int*)num1 < *(int*)num2) return -1;  
    if(*(int*)num1 == *(int*)num2) return 0;  
    if(*(int*)num1 > *(int*)num2) return 1;  
}
```

```
int main(){  
    int arr[N], bubble_arr[N];  
    int num = 1, exchng;  
    float bubble_time, qsort_time;  
  
    for(int i=0;i<N;i++){  
        scanf("%d", &arr[i]);  
        bubble_arr[i] = arr[i];  
    }  
  
    clock();  
    for(int i=0;i<N;i++){  
        num = 0;
```

```

        for(int j=0;j<N-i;j++){
            if(bubble_arr[j] > bubble_arr[j+1]){
                exchneg = bubble_arr[j+1];
                bubble_arr[j+1] = bubble_arr[j];
                bubble_arr[j] = exchneg;
                num = 1;
            }
        }
        if(num == 0)
            break;
    }
    bubble_time = clock()/CLOCKS_PER_SEC;

    clock();
    qsort(arr, N, sizeof(int), cmp);
    qsort_time = clock()/CLOCKS_PER_SEC;

    for(int i=0;i<N;i++)
        printf("%d ", arr[i]);
    printf("\n%f\n%f", bubble_time, qsort);

    return 0;
}

```