

**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. 1304

\_\_\_\_\_

Андреев В.В.

Преподаватель

\_\_\_\_\_

Чайка К. В.

Санкт-Петербург

2022

## Цель работы.

Разобраться со стандартной библиотекой языка Си на примере оценки времени работы быстрой сортировки при помощи стандартной библиотеки `time.h`.

## Задание.

Вариант 3.

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.

Заголовочные файлы: `stdio.h` `stdlib.h` `time.h`

Функции: Перечень функций представлен в табл. 1

Таблица 1 – Функции программы

Имя функции	Возвращаемое значение	Аргументы	Комментарии
CompareInts	Результат сравнения чисел. < 0, если первое число меньше второго 0, если числа равны > 0, если первое число больше второго	<code>const void* A</code> — указатель на первое число для сравнения <code>const void* B</code> — указатель на второе число для сравнения	Функция для сравнения чисел. Используется как аргумент в быстрой сортировке.
BubbleSort	—	<code>int* Arr</code> – указатель на массив чисел для сортировки <code>int ArrSize</code> – размер массива	Сортирует массив методом «Сортировка пузырьком».

Алгоритм работы:

1. В цикле считываем последовательность чисел и записываем их сразу в массивы `Arr1` и `Arr2`. (Получаем 2 одинаковых массива).
2. Записываем в переменную `start` количество тактов процессора с начала работы программы при помощи функции `clock()`.
3. Вызываем функцию `BubbleSort` для `Arr1`.
4. Записываем в переменную `end` количество тактов процессора с начала работы программы при помощи функции `clock()`.
5. Записываем в переменную `TimeForBubbleSort` время выполнения сортировки пузырьком.  
$$TimeForBubbleSort = (double)(stop - start) / CLOCKS\_PER\_SEC;$$
Где `CLOCKS_PER_SEC` – количество тактов процессора, совершаемых за одну секунду.
6. Записываем в переменную `start` количество тактов процессора с начала работы программы при помощи функции `clock()`.
7. Вызываем функцию `qsort` для `Arr2` при помощи функции `CompareInts`.  
$$qsort(Arr2, ArraySize, sizeof(int), CompareInts);$$

8. Записываем в переменную `end` количество тактов процессора с начала работы программы при помощи функции `clock()`.
9. Записываем в переменную `TimeForQuickSort` время выполнения быстрой сортировки.  

$$TimeForQuickSort = (double)(stop - start) / CLOCKS\_PER\_SEC;$$
10. Выводим в цикле отсортированный массив `Arr1`.
11. Выводим `TimeForBubbleSort` и `TimeForQuickSort`.

## Тестирование.

Результаты тестирования представлены в табл. 2.

Таблица 2 – Результаты тестирования

№ п/п	Входные данные	Выходные данные	Комментарии
1.	10 9 8 7 6 5 4 3 2 1	1 2 3 4 5 6 7 8 9 10	Ответ верный.
2.	9 6 7 1 2 3 0 8 7 55	0 1 2 3 6 7 7 8 9 55	Ответ верный.
3.	9 9 0 0 0 0 0 0 -9 -9	-9 -9 0 0 0 0 9 9	Ответ верный.

## Выводы.

Была изучена стандартная библиотека языка Си и полученные знания были применены для написания программы, сортирующей массив двумя способами – сортировка пузырьком и быстрая сортировка из стандартной библиотеки.

Для каждого способа программа выводит время выполнения алгоритма при помощи стандартной библиотеки `time.h`.

## ПРИЛОЖЕНИЕ А

### ИСХОДНЫЙ КОД ПРОГРАММЫ

Название файла: main.c

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

#define ArraySize 1000

int CompareInts(const void* A, const void* B)
{
    return *(int*)A - *(int*)B;
}

void BubbleSort(int* Arr, int ArrSize)
{
    int TmpMinElemIndex;
    int TmpVal;

    for(int i = 0; i < ArrSize; ++i)
    {
        TmpMinElemIndex = i;

        for(int j = i + 1; j < ArrSize; ++j)
        {
            if(Arr[j] < Arr[TmpMinElemIndex])
            {
                TmpMinElemIndex = j;
            }
        }

        TmpVal = Arr[i];
        Arr[i] = Arr[TmpMinElemIndex];
        Arr[TmpMinElemIndex] = TmpVal;
    }
}

int main()
{
    int Arr1[ArraySize];
    int Arr2[ArraySize];
    clock_t start, stop;

    for(int i = 0; i < ArraySize; ++i)
    {
        scanf("%i", Arr1 + i);
        Arr2[i] = Arr1[i];
    }
}
```

```

start = clock();
BubbleSort(Arr1, ArraySize);
stop = clock();
double TimeForBubbleSort = (double)(stop - start)/CLOCKS_PER_SEC;

start = clock();
qsort(Arr2, ArraySize, sizeof(int), CompareInts);
stop = clock();
double TimeForQuickSort = (double)(stop - start)/CLOCKS_PER_SEC;

for(int i = 0; i < ArraySize; ++i)
{
    printf("%i ", Arr1[i]);
}
printf("\n");

printf("%lf\n", TimeForBubbleSort);
printf("%lf", TimeForQuickSort);

return 0;
}

```