

**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

\_\_\_\_\_

Ефремов А.А.

Преподаватель

\_\_\_\_\_

Чайка К.В.

Санкт-Петербург

2022

### **Цель работы.**

Изучить стандартную библиотеку языка Си, научиться применять на практике библиотеки `time.h` и `stdlib.h`.

### **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.**

В функции `main` происходит считывание данных, заполнение двух массивов `arr` и `arr1`, замер времени для необходимых функций сортировки этих массивов при помощи `clock()`, а также последующий вывод результатов.

Функция `bsort()` реализует сортировку «пузырьком», принимая массив и проводя операции над его элементами.

Функция `cmp` реализует сравнение двух объектов массива для функции `qsort` из `main`.

Разработанный программный код см. в приложении А.

### Тестирование.

Результаты тестирования представлены в табл. 1.

Таблица 1 – Результаты тестирования

№ п/п	Входные данные	Выходные данные	Комментарии
1.	4 3 2 6 3 9 1 6 3 8	1 2 3 3 3 4 6 6 8 9 0.000002 0.000003	Тест проводился для размера массива, равного 10. Время указывается в секундах.
2.	6 3 44 7 66 2 76 3 67 35 18 0 28 1 35	0 1 2 3 3 6 7 18 28 35 35 44 66 67 76 0.000009 0.000010	Тест проводился для размера массива, равного 15. Время указывается в секундах.
3.	8 2 3 4 2 3 8 4 5 0 1 5 2 3 7 4 3 10 21 4 3 78 45 23 22	0 1 2 2 2 3 3 3 3 4 4 4 4 5 5 7 8 8 10 21 22 23 45 78 0.000013 0.000010	Тест проводился для размера массива, равного 25. Время указывается в секундах.

### Выводы.

Была написана программа, принимающая массив чисел, сортирующая их разными методами и выводящая в консоль отсортированный массив с подсчитанным временем, которое было уделено на сортировку.

Были изучены основные библиотеки языка Си и успешно применены на практике библиотеки `time.h` и `stdlib.h`.

## ПРИЛОЖЕНИЕ А

### ИСХОДНЫЙ КОД ПРОГРАММЫ

Название файла: lb1.c

```
#include <stdio.h>
#include <stdlib.h>
#include <time.h>
void bsort(int* arr, int N){
    for (int i = 0; i < N-1; i++){
        for (int j = 0; j < N - i - 1; j++){
            int p = arr[j];
            if (arr[j+1] < arr[j]){
                arr[j] = arr[j+1];
                arr[j+1] = p;
            }
        }
    }
}

int cmp(const void* a, const void* b){
    int f = *((int*)a);
    int s = *((int*)b);
    return f - s;
}

int main(){
    int N = 1000;
    clock_t bstart, qstart, bend, qend;
    int arr[N], arr1[N];
    for (int i = 0; i < N; i++){
        scanf("%i", &arr[i]);
        arr1[i] = arr[i];
    }
    bstart = clock();
    bsort(arr, N);
    bend = clock();
    float total_btime = (float)(bend-bstart)/CLOCKS_PER_SEC;

    qstart = clock();
    qsort(arr1, N, sizeof(int), cmp);
    qend = clock();
    float total_qtime = (float)(qend-qstart)/CLOCKS_PER_SEC;

    for (int i = 0; i < N; i++)
        printf("%i ", arr[i]);
    printf("\n%f\n", total_btime);
    printf("%f\n", total_qtime);
}
```