

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

Кривоченко Д. И.

Преподаватель

Чайка К. В.

Санкт-Петербург

2022

Цель работы.

Изучить стандартную библиотеку языка Си. Научиться применять функции, содержащиеся в ней.

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 начале программы объявляются 3 библиотеки: `stdio.h`, `stdlib.h`, `math.h`, `time.h`. Библиотека `time.h` используется для подсчёта времени работы программы.

В главной функции `main()` находится алгоритм сортировки пузырьком и вызов `qsort` для массива чисел (одинаковых `arr_bubble` и `arr_qsort` соотв.). Также замеряется время их работы с помощью `clock_t` и после выводится в консоль.

Выводы.

В ходе выполнения лабораторной работы были изучены функции стандартной библиотеки Си. Написана программа, сортирующая массив с помощью `qsort` и `bubble sort`, считающая время сортировки.

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 * a,
        const void * b) {
    const int * f = (int * ) a;
    const int * s = (int * ) b;
    if ( * f > * s)
        return 1;
    if ( * f < * s)
        return -1;
    return 0;
}

int main() {

    int arr_qsort[N];
    int arr_bubble[N];
    int q;
    for (int i = 0; i < N; i++) {
        scanf("%d", & arr_qsort[i]);
        arr_bubble[i] = arr_qsort[i];
    }

    clock_t time_start_bubble = clock();
    for (int i = 0; i < N - 1; i++) {
        for (int j = 0; j < N - i - 1; j++) {
```

```

        if (arr_bubble[j] > arr_bubble[j + 1]) {
            q = arr_bubble[j];
            arr_bubble[j] = arr_bubble[j + 1];
            arr_bubble[j + 1] = q;
        }
    }
}

clock_t time_end_bubble = clock() - time_start_bubble;

for (int i = 0; i < N; i++) {
    printf("%d ", arr_bubble[i]);
}
puts("");

clock_t time_start_qsort = clock();
qsort(arr_qsort, N, sizeof(int), cmp);
clock_t time_end_qsort = clock() - time_start_qsort;
    printf("%f\n", ((float) time_end_bubble) /
CLOCKS_PER_SEC);
    printf("%f\n", ((float) time_end_qsort) / CLOCKS_PER_SEC);
    return 0;

}

```