

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

ОТЧЕТ
по лабораторной работе №1
по дисциплине «Программирование»
Тема: Обзор стандартной библиотеки

Студент гр. 0382

Павлов С.Р

Преподаватель

Берленко Т.А

Санкт-Петербург

2021

Цель работы.

Изучить и освоить функционал стандартной библиотеки языка Си.

Задание.

Напишите программу, на вход которой подается массив целых чисел длины 1000.

Программа должна совершать следующие действия:

- посчитать время, за которое будет совершена сортировка, используя при этом функцию стандартной библиотеки
- отсортировать массив с помощью алгоритма "сортировка пузырьком"
- 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 главной функции *int main()*, инициализируются переменные *horal_1* и *horal_2* типа *float* , затем целочисленный массив *array* и переменная *i* типа *int*. Затем пользователю предлагается ввести массив 1000-и целочисленных чисел. Затем функция *float bubble_sort(int* array)* , производит сортировку пузырьком, данного массива, и возвращает время в секундах, на исполнение данной

сортировки, это реализуется с помощью функции *clock()*, начало и конец отчета времени соответственно записываются в переменные *start* и *end* типа *clock_t* их разница является кол-вом тактов для выполнения сортировки. Далее полученное кол-во тактов делится на *clocks_per_sec*, и затем функция возвращает число типа *float*, являющиеся кол-вом времени затраченным на сортировку в секундах.

Следующая функция быстрой сортировки *qsort*, реализуется с помощью функции стандартной библиотеки (*stdlib.h*), на вход функция получает полученный копию изначального не отсортированного массива, его размер, размер целочисленного типа, и функцию компаратора *qsort_compare(const void *pa, const void *pb)*. Далее с помощью выше описаного метода, измеряется время работы быстрой сортировки. Затем функция выводит отсортированный массив, затем время пузырьковой и быстрой сортировок в секундах.

Разработанный программный код смотрите в приложении А.

Тестирование.

Результаты тестирования представлены в табл. 1.

Таблица 1 — Результаты тестирования.

| | Входные данные | Выходные данные | Коментарии |
|----|---|--|-----------------------------|
| 1. | 827 888 915 488 (далее случайно сгенерирование остальные 1000 чисел) | 2 3 3 3 4 5 (далее отсортированный массив чисел) 0.002563 0.000092 | Программа работает верно |
| 3. | 261 84 934 896 929 551 457 396 624 668 | 261 84 934 896 929 551 457 396 624 668 0.000001 0.000001 | Программа работает верно |

Выводы.

Была изучена стандартная библиотека языка Си.

Разработана программа использующая функции библиотеки для сортировки и посчитывания времени сортировки.

Так же в ходе работы было выявлено значительное опережение в скорости алгоритма функции быстрой сортировки над сортировкой пузырьком.