

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

Крупин Н. С.

Преподаватель

Чайка К. В.

Санкт-Петербург

2022

Цель работы

Освоение работы с функциями стандартной библиотеки.

Задание

Вариант 4.

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

Программа должна совершать следующие действия:

- отсортировать массив по невозрастанию модулей элементов 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 быстрая сортировка.

Отсортированный массив, время быстрой сортировки должны быть выведены с новой строки, при этом элементы массива должны быть разделены пробелами».

Выполнение работы

В функции `main()` используются массив `arr` на 1000 целых чисел и переменная для хранения числа тактов работы процессора `clocks` типа `clock_t`. В функции выполняются следующие шаги:

- запись 1000 чисел с консоли в массив с помощью функции стандартной библиотеки `scanf()`;
- сохранение текущего количество тактов в `clocks` с помощью функции стандартной библиотеки `clock()`;
- сортировка массива с помощью функции стандартной библиотеки `qsort()` и функции-компаратора `cmp()`;
- перезапись в `clocks` разницы между прежним записанным и текущим числом тактов с помощью функции стандартной библиотеки `clock()`;
- перевод числа тактов в секунды с помощью константы `CLOCKS_PER_SEC` стандартной библиотеки и вывод результата в консоль с помощью функции стандартной библиотеки `printf()`.

Функция-компаратор `cmp()` принимает в качестве аргументов два указателя `arg1` и `arg2` типа `const void*`, приводит их к типу `int*` и записывает значения, на которые они указывают, в целочисленные переменные `x1` и `x2`. Функция сравнивает `x1` и `x2` с константой `INT_MIN` стандартной библиотеки и возвращает -1, 0, 1 или разность модулей `x2` и `x1` с помощью функции `abs()` стандартной библиотеки в зависимости от результата сравнения, используя функцию `abs()` только в случае, если ни одно число не равно `INT_MIN`, чтобы не вызвать предупреждение и гарантировать неотрицательность возвращаемого `abs()` значения. Таким образом, функция возвращает положительное число, если `x2` по модулю больше `x1` и должен находиться в отсортированном массиве левее, отрицательное число, если меньше и должен находиться правее, и ноль, если модули равны и порядок чисел не важен.

Разработанный программный код см. в приложении А.

Тестирование

Результаты тестирования для 4-байтного `int` представлены в таблице 1.

1. Тест функционала сортировки для 10 чисел.
2. Тест сортировки на предельных значениях типа `int` для 10 чисел.
3. Тест примерного времени сортировки для 1000 чисел от -499 до 500.
4. Тест примерного времени сортировки для 1000 случайных чисел в пределах от $-\text{INT_MAX}/2$ до $\text{INT_MAX}/2 - 1$.

Таблица 1 – Результаты тестирования

№ п/п	Входные данные	Выходные данные
1.	4 -7 0 6 -4 3 4 -1 0 5	-7 6 5 4 -4 4 3 -1 0 0 0.000011
2.	0 2147483647 -2147483648 -2147483648 -2147483648 -2147483647 2147483647 0 -2147483648 5	-2147483648 -2147483648 -2147483648 -2147483648 2147483647 -2147483647 2147483647 5 0 0 0.000012
3.	-499 -498 ... 498 499 500	500 -499 499 -498 498 ... -2 2 -1 1 0 0.000204
4.	<code>srand(time(NULL));</code> <code>for (int i = 0; i < N; i++)</code> <code>arr[i] = rand() % INT_MAX - INT_MAX/2;</code>	// 1000 упорядоченных чисел. // 0.000313

Выводы

Были изучены основы работы с функциями стандартной библиотеки языка C, использованы объявления из заголовочных файлов стандартной библиотеки: `stdio.h`, `stdlib.h`, `limits.h` и `time.h`.

Разработана программа, выполняющая сортировку введённого с консоли массива целых чисел, подсчёт времени сортировки и вывод результата в консоль.

Программа работает только для фиксированного количества введённых данных, удовлетворяющих типу `int`.

ПРИЛОЖЕНИЕ А

ИСХОДНЫЙ КОД ПРОГРАММЫ

Название файла: main.c

```
#include <stdio.h>
#include <stdlib.h>
#include <limits.h>
#include <time.h>
#define N 1000

int cmp(const void* arg1, const void* arg2){
    int x1 = *(int*)arg1;
    int x2 = *(int*)arg2;

    if (x1 == INT_MIN && x2 != INT_MIN) return -1;
    if (x1 == INT_MIN && x2 == INT_MIN) return 0;
    if (x1 != INT_MIN && x2 == INT_MIN) return 1;
    else return abs(x2) - abs(x1);
}

int main(){
    int arr[N];
    clock_t clocks;

    for (int i = 0; i < N; i++)
        scanf("%d", arr+i);

    clocks = clock();
    qsort(arr, N, sizeof(int), cmp);
    clocks = clock() - clocks;

    for (int i = 0; i < N; i++)
        printf("%d ", arr[i]);
    printf("\n%f\n", (double)clocks/CLOCKS_PER_SEC);

    return 0;
}
```