

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

Шаврин А.П

Преподаватель

Чайка К.В.

Санкт-Петербург

2022

Цель работы.

Изучить возможности стандартной библиотеки языка Си, более подробно разобраться в библиотеке *time.h*, чтобы получить практику работы со временем, а также изучить функции сортировки *qsort* и *bubble sort*, разобраться в алгоритмах сортировки.

Задание.

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.

Основные теоретические положения.

1. *void qsort (void *first, size_t number, size_t size, int (* comparator) (const void *, const void *));*

- *first*
Указатель на первый элемент сортируемого массива.
- *number*
Количество элементов в сортируемом массиве, на который ссылается указатель *first*.

- *size*
Размер одного элемента массива в байтах.
- *comparator*
Функция, которая сравнивает два элемента. Функция должна иметь следующий прототип:

```
int funccmp( const void * val1, const void * val2 );
```

```
1. clock_t clock (void);
```

Количество тактов, прошедших с эпохи, связанной с выполнением конкретной программы.

При сбое функция возвращает значение -1.

clock_t это тип, определенный в *<ctime>* качестве псевдонима фундаментального арифметического типа.

Выполнение работы.

- Поскольку функции сортировки преобразуют сам массив, была реализована функция *void read_arr(int arr1[], int arr2[])*, в которой происходит заполнение сразу 2х массивов (одинаковыми данными).
- Для вывода массива реализована функция *void print_arr(int arr[])*.
- Для сортировки пузырьком реализована функция *void bubble_sort(int arr[])*.
- Для сравнения элементов функцией *qsort* реализована функция *int cmp(const void *a, const void *b)* (компаратор).
- В функции *main* происходит инициализация переменных, заполнение массивов, замер времени для *bubble sort* и *qsort*, вывод результатов в соответствии с требованием.

Результаты тестирования см. в приложении Б.

Разработанный программный код см. в приложении А.

Выводы.

Были изучены основные библиотеки Си, в частности работа с библиотекой *time.h*, проанализировано время работы быстрой сортировки и сортировки пузырьком.

ПРИЛОЖЕНИЕ А

ИСХОДНЫЙ КОД ПРОГРАММЫ

Название файла: main.c

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

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

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

```
#define N 1000
```

```
void read_arr(int arr1[], int arr2[]){  
    for (int i = 0; i < N; i++){  
        scanf("%d", &arr1[i]);  
        arr2[i] = arr1[i];  
    }  
}
```

```
void print_arr(int arr[]){  
    for (int i = 0; i < N; i++){  
        printf("%d ", arr[i]);  
        printf("\n");  
    }  
}
```

```
void bubble_sort(int arr[]){  
    for (int i = N-1; i >= 0; i--){  
        int no_swap = 1;  
        for (int j = 0; j < i; j++){  
            if (arr[j] > arr[j+1]){  
                int tmp = arr[j];  
                arr[j] = arr[j+1];  
                arr[j+1] = tmp;  
                no_swap = 0;  
            }  
        }  
        if (no_swap == 1) break;  
    }  
}
```

```

        arr[j+1] = tmp;
        no_swap = 0;
    }
}
if (no_swap)
    break;
}
}

```

```

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

```

```

int main(){
    //init var
    int arr1[N];
    int arr2[N];
    clock_t tstart;
    clock_t tbsrt;
    clock_t tqsr;

    //read arr

```

```

    read_arr(arr1, arr2);

    //bubble sort
    tstart = clock();
    bubble_sort(arr1);
    tbsrt = clock() - tstart;

    //quick sort
    tstart = clock();
    qsort(arr2, N, sizeof(int), cmp);
    tqsrt = clock() - tstart;

    //print ans
    print_arr(arr1);
    printf("%f\n", (float)tbsrt/CLOCKS_PER_SEC);
    printf("%f\n", (float)tqsrt/CLOCKS_PER_SEC);
    return 0;
}

```

ПРИЛОЖЕНИЕ Б

ТЕСТИРОВАНИЕ

Таблица Б.2 - Примеры тестовых случаев

№ п/п	Входные данные	Выходные данные	Комментарии
1.	0 2 5 -39 -234 345 3 2 6 0	-234 -39 0 0 2 2 3 5 6 345 0.000003 0.000004	Время считается в секундах, для данного теста длина массива устанавливается в 10 элементов. Результат связан с реализацией функций.
2.	100 99 98 97 96 95 94 93 92 91 90 89 88 87 86 85 84 83 82 81 80 79 78 77 76 75 74 73 72 71 70 69 68 67 66 65 64 63 62 61 60 59 58 57 56 55 54 53 52 51 50 49 48 47 46 45 44 43 42 41 40 39 38 37 36 35 34 33 32 31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0 -1 -2 -3 -4 -5 -6 -7 -8 -9 -10 - 11 -12 -13 -14 -15 -16 -17 - 18 -19 -20 -21 -22 -23 -24 - 25 -26 -27 -28 -29 -30 -31 - 32 -33 -34 -35 -36 -37 -38 - 39 -40 -41 -42 -43 -44 -45 - 46 -47 -48 -49 -50 -51 -52 - 53 -54 -55 -56 -57 -58 -59 - 60 -61 -62 -63 -64 -65 -66 - 67 -68 -69 -70 -71 -72 -73 - 74 -75 -76 -77 -78 -79 -80 - 81 -82 -83 -84 -85 -86 -87 - 88 -89 -90 -91 -92 -93 -94 -	-100 -99 -98 -97 -96 -95 -94 -93 -92 -91 -90 -89 -88 -87 -86 -85 -84 -83 -82 -81 -80 -79 -78 -77 -76 -75 -74 -73 -72 -71 -70 -69 -68 -67 -66 -65 -64 -63 -62 -61 -60 -59 -58 -57 -56 -55 -54 -53 -52 -51 -50 -49 -48 -47 -46 -45 -44 -43 -42 -41 -40 -39 -38 -37 -36 -35 -34 -33 -32 -31 -30 -29 -28 -27 -26 -25 -24 -23 -22 -21 -20 -19 -18 -17 -16 -15 -14 -13 -12 -11 -10 -9 -8 -7 -6 -5 -4 -3 - 2 -1 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91	Время считается в секундах, для данного теста длина массива устанавливается в 201 элемент.

	95 -96 -97 -98 -99 -100	92 93 94 95 96 97 98 99 100 0.000234 0.000017	
3.	-100 -99 -98 -97 -96 -95 -94 -93 -92 -91 -90 -89 -88 -87 -86 -85 -84 -83 -82 -81 -80 -79 -78 -77 -76 -75 -74 -73 -72 -71 -70 -69 -68 -67 -66 -65 -64 -63 -62 -61 -60 -59 -58 -57 -56 -55 -54 -53 -52 -51 -50 -49 -48 -47 -46 -45 -44 -43 -42 -41 -40 -39 -38 -37 -36 -35 -34 -33 -32 -31 -30 -29 -28 -27 -26 -25 -24 -23 -22 -21 -20 -19 -18 -17 -16 -15 -14 -13 -12 -11 -10 -9 -8 -7 -6 -5 -4 -3 -2 -1 0 1 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100	-100 -99 -98 -97 -96 -95 -94 -93 -92 -91 -90 -89 -88 -87 -86 -85 -84 -83 -82 -81 -80 -79 -78 -77 -76 -75 -74 -73 -72 -71 -70 -69 -68 -67 -66 -65 -64 -63 -62 -61 -60 -59 -58 -57 -56 -55 -54 -53 -52 -51 -50 -49 -48 -47 -46 -45 -44 -43 -42 -41 -40 -39 -38 -37 -36 -35 -34 -33 -32 -31 -30 -29 -28 -27 -26 -25 -24 -23 -22 -21 -20 -19 -18 -17 -16 -15 -14 -13 -12 -11 -10 -9 -8 -7 -6 -5 -4 -3 - 2 -1 0 1 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 0.000003 0.000018	Время считается в секундах, для данного теста длина массива устанавливается в 201 элемент. Результат связан с реализацией функций.