

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

ОТЧЕТ
по лабораторной работе №4
по дисциплине «Программирование»
ТЕМА: СТРУКТУРЫ И ОБЗОР STL

Студент гр.0382

Диденко Д.В.

Преподаватель

Жангиров Т.Р.

Санкт-Петербург

2021

Цель работы.

Изучение правил работы со структурами и объединениями и возможностей стандартной библиотеки.

Задание.

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.

Основные теоретические положения.

Структура — это набор из одной или более переменных, возможно различных типов, сгруппированных под одним именем для удобства обработки.

Описание структуры, состоящее из заключенного в фигурные скобки списка описаний, начинается с ключевого слова STRUCT. За словом STRUCT может следовать необязательное имя, называемое ярлыком структуры. Такой ярлык именует структуры этого вида и может использоваться в дальнейшем как сокращенная запись подробного описания.

Элементы или переменные, упомянутые в структуре, называются членами. Ярлыки и члены структур могут иметь такие же имена, что и

обычные переменные (т.е. Не являющиеся членами структур), поскольку их имена всегда можно различить по контексту.

Член определенной структуры может быть указан в выражении с помощью конструкции вида : имя структуры . Член

time.h

В заголовочном файле time.h можно найти объявления типов и функций для работы с датой и временем. В том числе:

- Функция, позволяющая получить текущее календарное время
- Функция, позволяющая получить время в тактах процессора с начала выполнения программы
- Функция для вычисления разности в секундах между двумя временными штампами
- Функции для вывода значения даты и времени на экран

А также структура tm, содержащая компоненты календарного времени и функция для преобразования значения времени в секундах в объект такого типа.

assert.h

В стандартной библиотеке объявлен макрос препроцессора assert(). С его помощью можно выполнять проверку некоторых условий в процессе выполнения программы.

Если условие ложно, то в процессе выполнения будет выведена некоторая информация о том в каком месте это произошло (имя файла с исходным кодом, имя функции, номер строки) и само условие. Стоит заметить, что для отключения проверок достаточно всего лишь добавить макрос #define NDEBUG перед включением заголовочного файла assert.h

Использование assert может сильно упростить отладку программ, гарантируя, что все необходимые условия в процессе выполнения программы соблюдены.

stdarg.h

В языке Си возможно использование функций с произвольным числом и типом аргументов (вспомним функцию `printf`, например). Для удобства работы с такими аргументами есть средства, находящиеся в заголовочном файле `stdarg.h`

Заголовочный файл определяет специальный тип `va_list` и набор функций для работы с этим типом `va_start`, `va_arg`, `va_end`, с помощью которых можно по очереди перебирать аргументы функции.

ctype.h

В состав стандартной библиотеки входят функции для работы с символами, объявленные в заголовочном файле `ctype.h`

Эти функции позволяют переводить символы из верхнего регистра в нижний, определять является ли этот символ буквой, цифрой или каким-то служебным.

string.h

Стандартная библиотека Си содержит различные функции для работы с памятью и строками, объявленные в заголовочном файле `string.h`

Среди них есть функции для:

- копирования памяти
- сравнения памяти
- сравнения строк
- разбиения строки на токены
- конкатенации строк
- поиска символов и подстроки в строке
- определения длины строки

и некоторые другие

Для работы с широкими символами в программах на языке C используется тип `wchar_t`. Важно понимать, что фактическое представление символов в `wchar_t` не регламентировано и может отличаться в зависимости от платформы.

В ОС Linux `wchar_t` обычно имеет размер 4 байта и хранит символы в соответствии с UTF-32. Для использования типа `wchar_t`, требуется подключить заголовочный файл `wchar.h`.

Для работы с `wchar_t` существует специальный набор функций, аналогичный тому, который используется для строк и символов `char`, включая функции ввода/вывода. Функции для строк содержатся в заголовочном файле `wchar.h`, а для символов в файле `wctype.h`. (!)Важное замечание: для считывания кириллических символов с клавиатуры и вывода их на экран при использовании `wchar_t`, следует установить локаль по умолчанию, с помощью вызова функции `setlocale`.

Для этого следует подключить заголовочный файл `locale.h`.

В этом заголовочном файле собраны объявления различных функций:

- Функции для работы с динамической памятью
- Функции для преобразования строки в число
- Генерации псевдослучайных чисел
- Функции для управления процессом выполнения программы
- Функции для вычисления абсолютного значения и деления целых чисел
- Функции для сортировки и поиска

На последние 2 функции стоит обратить особое внимание: эти функции позволяют производить быструю сортировку(`qsort`) и бинарный поиск(`bsearch`) в массиве данных любого типа.

Выполнение работы.

Исходный код решения задачи см.в приложении А.

Подключаются библиотеки `time.h`, `stdlib.h`, `stdio.h`. С помощью функции `clock()` библиотеки `time.h` считывается текущее время работы программы и записывается в переменную `t`. Далее производится сортировка введенного

массива по невозрастанию с помощью функции `qsort()` библиотеки `stdlib.h`, в которой используется функция-компаратор `compare()`. После выполнения сортировки еще раз записывается время работы программы в переменную `t2`. Отсортированный массив и время сортировки — разница между `t2` и `t` выводится на экран.

Тестирование.

Результаты тестирования представлены в табл. 1.

*для проверки использовались примеры с 10 элементами в массиве

Таблица 1 – Результаты тестирования

№ п/п	Входные данные	Выходные данные	Комментарии
1.	5 2 3 6 8 1 4 9 5 6	9 8 6 6 5 5 4 3 2 1 5	Программа работает верно
2.	16 8 12 73 51 26 14 42 32 99	99 73 51 42 32 26 16 14 12 8 4	Программа работает верно

Выводы.

Были изучены принципы работы с указателя и массивами.

Разработана программа, которая форматирует некоторый введенный текст и выводит результат на консоль. При вводе предложения записываются в массивы, для которых была динамически выделена память. Текст хранится в виде двумерного динамического массива, элементы которого — указатели на массивы предложений.

ПРИЛОЖЕНИЕ А

ИСХОДНЫЙ КОД ПРОГРАММЫ

Название файла: main.c

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

#define N 1000

int compare(int* a,int* b){
    return abs(*a)<=abs(*b);
}

int main() {
    int digit[N];
    for (int i = 0;i<N;i++){
        scanf("%d",&digit[i]);
    }
    int t = clock();
    qsort(&digit[0],N,sizeof(int),(int (*)(const void*,const
void*))compare);
    int t2 = clock();
    for(int j = 0;j<N;j++){
        printf("%d ",digit[j]);
    }
    printf("\n%d",t2-t);
    return 0;
}
```