

МИНОБРНАУКИ РОССИИ
Санкт-Петербургский государственный
электротехнический университет
«ЛЭТИ» им. В.И. Ульянова (Ленина)
Кафедра МОЭВМ

ОТЧЕТ

по лабораторной работе №2
по дисциплине «Программирование»
Тема: Условия, циклы, оператор switch

Студент(ка) гр. 7381

Алясова А.Н.

Преподаватель

Берленко Т.А.

Санкт-Петербург

2017

Цель работы:

Научиться пользоваться оператором выбора Switch и условным оператором While. Отточить навыки создания Makefile.

Задание:

В текущей директории создайте проект с make-файлом. Главная цель должна приводить к сборке проекта. Файл, который реализует главную функцию, должен называться **menu.c**; исполняемый файл - **menu**. Определение каждой функции должно быть расположено в отдельном файле, название файлов указано в скобках около описания каждой функции.

Реализуйте функцию-меню, на вход которой подается одно из значений 0, 1, 2, 3 и массив целых чисел размера не больше 100. Числа разделены пробелами. Строка заканчивается символом перевода строки.

В зависимости от значения, функция должна выводить следующее:

0 : максимальное число в массиве. (**max.c**)

1 : минимальное число в массиве. (**min.c**)

2 : разницу между максимальным и минимальным элементом. (**diff.c**)

3 : сумму элементов массива, расположенных до минимального элемента. (**sum.c**)

иначе необходимо вывести строку "Данные некорректны".

Основные теоретические положения:

В данной лабораторной работе создан файл **menu.c**. В этой функции пользователю необходимо вводить номер операции и массив целых чисел с помощью функции `scanf()`, основная функция ввода с консоли. Она предназначена для ввода данных любого встроенного типа и автоматически преобразует введенное число в заданный формат. Прототип из файла `STDIN.H`.

Введенный номер операции поступает в функцию множественного выбора `switch`:

Оператор множественного выбора

```
switch (<выражение>)  
{  
    case <константное выражение 1>: <операторы 1>  
        ...  
    case <константное выражение N>: <операторы N>  
        [default: <операторы>]  
}
```

Оператор множественного выбора `switch` выполняет поочередное сравнение выражения со списком константных выражений. При совпадении, выполнение программы начинается с соответствующего оператора. В случае, если совпадений не было, выполняется необязательная ветка **default**. Важно помнить, что операторы после первого совпадения будут выполняться далее один за другим. Чтобы этого избежать, следует использовать оператор **break**, который досрочно прерывает выполнение цикла.

Для того чтобы заголовочные файлы не подключались только один раз использовался код:

```
#ifndef <name>
#define <name>
<type> <name>
#endif
```

Где **#ifndef**, **#define**, **#endif** – директивы, которые включает в себя препроцессор.

Директива **#define** вводит макроопределение или макрос. Общая форма директивы следующая:

***#define** имя_макроса последовательность_символов*

Последовательность символов называют еще строкой замещения. Когда препроцессор находит в исходном тексте программы *имя_макроса*, он заменяет его на *последовательность_символов*. Любые вхождения макроса, обнаруженные в строках символов, символьных константах или в комментариях, замене не подлежат.

Директивы **#endif** и **#ifndef** – директивы условной компиляции. Если выражение следующее за **#ifndef**, истинно, то коды, заключенные между **#endif** и **#ifndef**, будут компилироваться. Выражение следующие за **#ifndef**, проверяется во время компиляции, поэтому оно может содержать только константы и макросы, которые прежде определены. Переменные не могут использоваться.

В функции **diff.c** вызываются функции **max.c** и **min.c**, а в функции **sum.c** вызывается функция **min.c**, для того чтобы избежать ошибки дублирования кода. Далее создаётся **Makefile** в котором компилируется программа; исполняемый файл **menu.c** обретает имя **menu** с помощью ключа “-o”.

Вывод:

В данной лабораторной работе изучены: оператор выбора (switch) и условный оператор(while). Также отточены навыки создания Makefile. Изучен ввод массива пользователем с помощью (scanf()), вызов функций из файлов.

Исходный код программы:

Makefile

```
all: menu.o max.o min.o diff.o sum.o
    gcc menu.o max.o min.o diff.o sum.o -o menu
menu.o: menu.c max.h min.h diff.h sum.h
    gcc -c menu.c
max.o: max.c max.h
    gcc -c max.c
min.o: min.c min.h
```

```
gcc -c min.c
diff.o: diff.c diff.h max.h min.h
gcc -c diff.c
sum.o: sum.c sum.h min.h
gcc -c sum.c
```

menu.c

```
#include <stdio.h>
#include "max.h"
#include "min.h"
#include "diff.h"
#include "sum.h"

int main() {

    int i=0;
    int n , result;
    char c;
    int arr[100];
    scanf("%d%c", &n, &c);
    scanf("%d%c", &arr[0], &c);
    while (c!='\n') {
        i++;
        scanf("%d%c", &arr[i], &c);
    }
    i++;
    result = 0;
    switch(n){
    case 0: result = max(arr, i);
        printf("%d\n", result );
        break;
    case 1: result = min(arr, i);
        printf("%d\n", result);
        break;
    case 2: result = diff(arr, i);
        printf("%d\n", result);
        break;
    case 3: result = sum(arr, i);
        printf("%d\n", result);
        break;
    default: printf("Данные некоректны\n");
    }
}
```