

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: УСЛОВИЯ, ЦИКЛЫ, ОПЕРАТОР SWITCH

Студентка гр. 1304

Ярусова Т. В.

Преподаватель

Чайка К. В.

Санкт-Петербург

2021

Цель работы.

Научиться использовать условия, циклы и оператор switch для написания простейших программ на языке Си.

Задание.

Напишите программу, выделив каждую подзадачу в отдельную функцию.

Реализуйте программу, 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 по модулю число в массиве. (abs_max)

1 : минимальное по модулю число в массиве. (abs_min)

2 : разницу между максимальным по модулю и минимальным по модулю элементом. (diff)

3 : сумму элементов массива, расположенных после максимального по модулю элемента (включая этот элемент). (sum)

иначе необходимо вывести строку "Данные некорректны".

Основные теоретические положения.

Помимо stdio.h заголовочного файла стандартной библиотеки языка Си нужно использовать в заголовочном файле stdlib.h стандартной библиотеки языка Си для использования функции нахождения модуля числа.

Выполнение работы.

В главной функции *int main()* объявляем переменную *znachenie*, которая будет хранить в себе значение для оператора *switch*. Считываем значение в переменную *znachenie* с помощью функции *scanf*. Объявляем массив *arr[100]*, который будет хранить введенные значения, строковую переменную *s*, которая будет в себе хранить знак после числа(знак пробела или переноса строки), переменную *index*, в которой в будущем будет храниться последний индекс массива.

С помощью цикла *for* последовательно заполняем массив *arr*. Если встречается символ переноса строки, цикл прерывается оператором *break*. В переменную *index* сохраняется номер последней итерации. Вызываем оператор *switch*, передаем ему значение переменной *znachenie*. В зависимости от переданного значения описываем блоки оператора: *case 0*: с помощью функции *printf* выводит значение функции *abs_max()*, которая принимает на вход массив *arr[100]* и переменную *index*, и возвращает максимальное по модулю число в массиве, потом прерываем блок с помощью оператора *break*; *case 1*: с помощью функции *printf* выводит значение функции *abs_min()*, которая принимает на вход массив *arr[100]* и переменную *index*, и возвращает минимальное по модулю число в массиве, потом прерываем блок с помощью оператора *break*; *case 2*: с помощью функции *printf* выводит значение функции *diff()*, которая принимает на вход массив *arr[100]* и переменную *index*, и возвращает разницу между максимальным по модулю и минимальным по модулю элементом, потом прерываем блок с помощью оператора *break*; *case 3*: с помощью функции *printf* выводит значение функции *sum()*, которая принимает на вход массив *arr[100]* и переменную *index*, и возвращает сумму элементов массива, расположенных после максимального по модулю элемента (включая этот элемент) , потом прерываем блок с помощью оператора *break*; *default*: этот блок сработает в том случае, если ни одно из других зна-

чений не совпало, с помощью функции *printf* выводит “Данные некорректны”.

При выводе результата с помощью функции *printf*(“\n”) не забываем использовать символ переноса строки(примечание в условии).

Функцией *return 0* возвращаем 0, программа завершается.

Функции:

Функция *int abs_max(int arr[100],int index)* принимает на вход в качестве аргументов массив целых чисел *arr[100]* и переменную *index*, которая хранит в себе индекс последнего элемента массива. В функции объявляем переменную *int max* и присваиваем ей модуль значения первого элемента массива *abs(arr[0])*. С помощью цикла *for(int i = 1; i <= index; i++)* проходим по каждому элементу массива и с помощью условного оператора *if* сравниваем модуль каждого элемент с модулем максимального значения, который хранится в переменной *max*. Если элемент массива по модулю больше, чем значение переменной *max* по модулю, то присваиваем значение элемента данной переменной по модулю. Функцией *return max* возвращаем значение переменной *max*, то есть максимальное значение массива *arr[100]* по модулю.

Функция *int abs_min(int arr[100],int index)* принимает на вход в качестве аргументов массив целых чисел *arr[100]* и переменную *index*, которая хранит в себе индекс последнего элемента массива. В функции объявляем переменную *int min* и присваиваем ей модуль значения первого элемента массива *abs(arr[0])*. С помощью цикла *for(int i = 1; i <= index; i++)* проходим по каждому элементу массива и с помощью условного оператора *if* сравниваем модуль каждого элемент с модулем минимального значения, который хранится в переменной *min*. Если элемент массива по модулю меньше, чем значение переменной *min* по модулю, то присваиваем значение элемента данной переменной по модулю. Функцией *return min* возвращаем значение переменной *min*, то есть минимальное значение массива *arr[100]* по модулю.

Функция *int diff(int arr[100],int index)* принимает на вход в качестве аргументов массив целых чисел *arr[100]* и переменную *index*, которая хранит в себе индекс последнего элемента массива. Функцией *return (abs_max(arr,index) - abs_min(arr,index))* возвращаем разницу между максимальным по модулю значению и минимальным по модулю значению массива, вызывая вышеописанные функции *abs_max(arr,index)* и *abs_min(arr,index)*.

Функция *int sum(int arr[100],int index)* принимает на вход в качестве аргументов массив целых чисел *arr[100]* и переменную *index*, которая хранит в себе индекс последнего элемента массива. В функции объявляем переменную *int index_max*, в которой мы будем хранить индекс максимального по модулю значения массива. Объявляем переменную *int max*, которой присваиваем максимальное по модулю значения массива, вызывая вышеописанную функцию *int abs_max()*. С помощью цикла *for(int i = 1; i <= index; i++)* проходим по каждому элементу массива и с помощью условного оператора *if* сравниваем каждый элемент массива с максимальным по модулю значением, который хранится в переменной *max*. Если элемент массива равен значению переменной *max*, то присваиваем номер итерации переменной *index_max* и прерываем цикл оператором *break*. Объявляем переменную *int summa* и присваиваем ей 0. В данной переменной в будущем будет храниться сумма всех элементов массива *arr[100]*, начиная с максимального по модулю значения. С помощью цикла *for(int j = index_max; i <= index; i++)* проходим по каждому элементу массива и накапливаем их в переменной *summa*. Функцией *return summa* возвращаем сумму всех элементов массива *arr[100]*, начиная с максимального по модулю значения.

Разработанный программный код см. в приложении А.

Тестирование.

Результаты тестирования представлены в табл. 1.

Таблица 1 – Результаты тестирования

№ п/п	Входные данные	Выходные данные	Комментарии
1	0 -28 26 30 22 -13 -28 3 -12 8 10 -19 -26 11 -6 -18 -3 -2 -26 18 8 -19 -17 -11 -12 -23 19 -16 -11 9	30	Вызывается блок с функцией <i>int abs_max()</i> , которая возвращает максимальное по модулю значение массива <i>arr[100]</i>
2	1 -28 26 30 22 -13 -28 3 -12 8 10 -19 -26 11 -6 -18 -3 -2 -26 18 8 -19 -17 -11 -12 -23 19 -16 -11 9	-2	Вызывается блок с функцией <i>int abs_min()</i> , которая возвращает минимальное по модулю значение массива <i>arr[100]</i>
3	2 -28 26 30 22 -13 -28 3 -12 8 10 -19 -26 11 -6 -18 -3 -2 -26 18 8 -19 -17 -11 -12 -23 19 -16 -11 9	32	Вызывается блок с функцией <i>int diff()</i> , которая возвращает разницу между максимальным по модулю значение массива <i>arr[100]</i> и минимальным по модулю значению этого же массива
4	3 -28 26 30 22 -13 -28 3 -12 8 10 -19 -26 11 -6 -18 -3 -2 -26 18 8 -19 -17 -11 -12 -23 19 -16 -11 9	-124	Вызывается блок с функцией <i>int sum()</i> , которая возвращает сумму чисел массива <i>arr[100]</i> , начиная с максимального по модулю значения массива
5	4 -28 26 30 22 -13 -28 3 -12 8 10 -19 -26 11 -6 -18 -3 -2 -26 18 8 -19 -17 -11 -12 -23 19 -16 -11 9	Данные некор- ректны	Ни одно из значений не совпало, поэтому вызывается блок <i>default</i>

Выводы.

Я научилась использовать условия, циклы и оператор `switch` и смогла написать простейшую программу на языке Си.

Были изучены основные управляющие конструкции языка такие как оператор *if*, *switch* и цикл *for*.

Разработана программа, выполняющая считывание с клавиатуры исходных данных и команды пользователя. Для обработки команд пользователя использовались условные операторы *if*, оператор `switch` с четырьмя блоками `case` и блоком `default`, который сообщал об ошибке и некорректности введенных данных. Для выполнения блоков были написаны отдельные функции выполняющие команды, требуемые условием.

ПРИЛОЖЕНИЕ А

ИСХОДНЫЙ КОД ПРОГРАММЫ

Название файла: Yarusova_Tatyana_lb1/main.c

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

int abs_max(int arr[100],int index){
    int max = abs(arr[0]);
    for(int i = 1; i <= index; i++){
        if (abs(arr[i]) > abs(max)){
            max = arr[i];
        }
    }
    return max;
}

int abs_min(int arr[100], int index){
    int min = arr[0];
    for(int i = 1; i <= index; i++){
        if (abs(arr[i]) < abs(min)){
            min = arr[i];
        }
    }
    return min;
}

int diff(int arr[100], int index){
    return (abs_max(arr,index) - abs_min(arr,index));
}

int sum(int arr[100],int index){
    int index_max;
    int max = abs_max(arr,index);
    for(int i = 0; i <= index;i++){
        if (max == arr[i]){
            index_max = i;
            break;
        }
    }
    int summa = 0;
    for (int j = index_max; j <= index; j++){
        summa += arr[j];
    }
    return summa;
}

int main(){
    int znachenie;
    char c;
    scanf("%d", &znachenie);
    int arr[100];
    int index;

    for(int i = 0; i < 100; i++){
        scanf("%d%c", &arr[i],&c);
    }
}
```



```

        if (c == '\n'){
            index = i;
            break;
        }
    }
    switch (znachenie){
        case 0:
            printf("%d\n", abs_max(arr, index));
            break;
        case 1:
            printf("%d\n", abs_min(arr, index));
            break;
        case 2:
            printf("%d\n", diff(arr, index));
            break;
        case 3:
            printf("%d\n", sum(arr, index));
            break;
        default:
            printf("Данные некорректны\n");
    }
    return 0;
}

```