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

Студент гр. 1382 _____ Коренев Д.А.

Преподаватель _____ Жангиров Т.Г.

Санкт-Петербург
2021

Цель работы.

Проверить способность работать с условиями, циклами и оператором switch на языке C(Си).

Задание (Вариант 4).

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 массиве есть хотя бы один четный и нечетный элемент.

В зависимости от значения, функция должна выводить следующее:

0 : индекс первого чётного элемента. (index_first_even)

1 : индекс последнего нечётного элемента. (index_last_odd)

2 : Найти сумму модулей элементов массива, расположенных от первого чётного элемента и до последнего нечётного, включая первый и не включая последний. (sum_between_even_odd)

3 : Найти сумму модулей элементов массива, расположенных до первого чётного элемента (не включая элемент) и после последнего нечётного (включая элемент). (sum_before_even_and_after_odd)

иначе необходимо вывести строку "Данные некорректны".

Выполнение работы.

Для решения задачи были сделаны 4 функции:

1 — index_first_even — находит индекс первого четного элемента в данном нам массиве. Принимает на вход массив(int arr[]) , а так же длину (int len) — количество элементов этого массива, среди которых надо найти четный

элемент. Она возвращает индекс первого четного элемента, который она находит путем перебора элементов массива, пока не встретить искомое, циклом for.

2 — index_last_odd — находит индекс последнего нечетного элемента в данном массиве. Принимает на вход массив (int arr[]) , а так же длину (int len) — количество элементов этого массива, среди которых надо найти последний нечетный элемент. Она находит путем перебора всех элементов массива, присваивая к переменной odd индекс нечетного элемента, каждый раз, когда находит такой в массиве. После завершения цикла for, переменная odd хранит в себе значение индекса последнего нечетного элемента, которое возвращает функция.

3 — sum_between_even_odd — находит сумму модулей элементов массива, расположенных от первого чётного элемента и до последнего нечётного, включая первый и не включая последний. Принимает на вход массив (int arr[]), а так же длину (int len) — количество потенциальных элементов среди которых надо найти сумму их модулей. Внутри функции вызываются функции index_first_even и index_last_odd, принцип которых описан выше, и присваивает возвращаемые значения в переменные if_even и il_odd соответственно. Используя цикл for суммирует модули элементы массива начиная от элемента с индексом равным значению if_even и заканчивая элементом массива с индексом il_odd не включительно. Возвращает искомую сумму.

4 - sum_before_even_and_after_odd — находит сумму модулей элементов массива, расположенных до первого чётного элемента (не включая элемент) и после последнего нечётного (включая элемент). Принимает на вход массив (int arr[]), а так же длину (int len) — количество потенциальных элементов среди которых надо найти сумму модулей этих элементов. Внутри функции вызываются функции index_first_even и index_last_odd, принцип которых описан выше, и присваивает возвращаемые значения в переменные if_even и il_odd соответственно. Используются два цикла for: первый - суммирует модули элементов массива начиная с элемента под нулевым индексом (т.е. Первый в

массиве) и заканчивая элементом с индексом равным значению в переменной If_even не включительно, второй - суммирует модули элементов массива начиная с элемента под индексом равным значению в переменной il_odd и заканчивая последним (т.е. элемент под индексом равным значению переменной len). Возвращает искомую сумму.

В главной функции main объявляются необходимые в дальнейшем переменные: task, ans, c, len, arr. При помощи scanf считывается значение: 0, 1, 2 или 3, и присваивается переменной task, в зависимости от которого будет вызываться та или иная функция. При помощи цикла for считываются данные, которые будут помещены в массив arr. Каждую итерацию к переменная len увеличивается на единицу, а при помощи scanf элемент массива arr с индексом i и переменная с равны некоторому значению и знаку табуляции после него соответственно во входных данных . Если знак после значения — перенос строки, то цикл for досрочно прерывается при помощи оператора break. Далее используется оператор switch по значению task. Если task равен 0, выводится значение возвращаемое функцией index_first_even, если 1, выводится значение возвращаемое функцией index_last_odd, если 2, выводится значение возвращаемое функцией sum_between_even_odd, если 3, выводится значение возвращаемое функцией sum_before_even_and_after_odd, в случае по умолчанию, т.е. если значение task не было равно ни 0, ни 1, ни 2, ни 3, выводится строчка "Данные некорректны". На этом программа завершает работу.

Тестирование.

Результаты тестирования представлены в табл. 1.

Таблица 1 — Результаты тестирования

№ п/п	Входные данные	Выходные данные	Комментарий
1	0 1 13 -3 -16 11 4	3	0 — означает, что программа должна вывести первое четное число из массива [1 13 -3 -16 11 4]. -16 находится под 3 индексом в массиве и

			является первым четным числом. Ответ: 3
2	1 12 13 -1 -4 1 0 5		1 — означает, что программа должна вывести последнее нечетное числа из массив [12 13 -1 -4 1 0 5]. 5 — последнее нечетное число под индексом 6. Ответ: 6
3	2 13 -7 1 4 14 13 8 -5 -2 7 2 0 -6	46	2 — означает, что программа должна вызвать функцию sum_between_even_odd. Первый четный элемент — 4, последний нечетный — 7, сумма элементов между ними (4+14+13+8+5+2) равна 46. Ответ 46
4	3 13 -7 2 5 2 3 5 3 2 4 12	41	3 — означает, что программа должна вызвать функцию sum_before_even_and_after_odd. Первый четный элемент — 2, последний нечетный — 3. Сумма элементов равна (13+7+3+2+4+12) 41. Ответ: 41
5	12 4 2 -3 23 3 4	Данные некорректны	Для значения 12 не вызывается ни одна из 4х описанных функций, значит программа должна вывести «Данные некорректны»

Выводы.

Была изучена работа условий, циклов, оператора switch на языке С(Си). Также была изучена работа с массивами.

ПРИЛОЖЕНИЕ А

ИСХОДНЫЙ КОД ПРОГРАММЫ

Название программы: PR_Korenev_Danil_lb1.c

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

// 0
int index_first_even(int arr[], int len) {
    for (int i = 0; i < len; ++i) {
        if ((arr[i] % 2) == 0) {
            return (i);
        }
    }
}

// 1
int index_last_odd(int arr[], int len){
    int odd = -1;
    for (int i = 0; i < len; ++i){
        if ((arr[i] % 2) != 0){
            odd = i;
        }
    }
    return(odd);
}

// 2
int sum_between_even_odd(int arr[], int len){
    int sum = 0;
    int if_even = index_first_even(arr, len);
    int il_odd = index_last_odd(arr, len);

    for (int i = if_even; i < il_odd; ++i){
        sum+=abs(arr[i]);
    }

    return(sum);
}


// 3
int sum_before_even_and_after_odd(int arr[], int len){
    int sum = 0;
    int if_even = index_first_even(arr, len);
    int il_odd = index_last_odd(arr, len);

    for (int i = 0; i < if_even; ++i) {
        sum += abs(arr[i]);
    }

    for (int i = il_odd; i < len; ++i) {
        sum += abs(arr[i]);
    }
    return(sum);
}


int main()
```

```
{
    int task;
    int ans;
    char c;
    int len = 0;
    int arr[100];

    scanf("%d", &task);

    for (int i = 0; i < 100; ++i){
        scanf("%d%c", &arr[i], &c);
        len ++;
        if (c == '\n'){
            break;
        }
    }

    switch (task)
    {
        case 0:
            ans = index_first_even(arr, len);
            printf("%d\n", ans);
            break;
        case 1:
            ans = index_last_odd(arr, len);
            printf("%d\n", ans);
            break;
        case 2:
            ans = sum_between_even_odd(arr, len);
            printf("%d\n", ans);
            break;
        case 3:
            ans = sum_before_even_and_after_odd(arr, len);
            printf("%d\n", ans);
            break;
        default:
            printf("Данные некорректны");
            break;
    }

    return 0;
}
```