

**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. 0382

Крючков А.М.

Преподаватель

Жангиров Т.Р.

Санкт-Петербург

2020

### **Цель работы.**

Изучить условия, циклы, оператор switch в языке программирования C.  
Научиться работать с функциями.

### **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 : 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.**

Написание кода производилось на базе системы Linux Ubuntu 20.04 через интегрированную среду разработки Visual Studio Code.

Описание вводимых переменных:

- n — число, номер исполняемой функции. Вводится пользователем
- list - массив типа `int` размером 100, предназначенный для хранения целых чисел, введенных пользователем

- `listsize` — число типа `int`, хранит в себе количество записанных элементов массива `list`

Описание используемых функций:

- `index_first_even` (на вход массив типа `int` и размер этого массива типа `int`) — возвращает индекс первого чётного элемента.
- `index_last_odd` (на вход массив типа `int` и размер этого массива типа `int`) — возвращает индекс последнего нечётного элемента.
- `sum_between_even_odd` (на вход массив типа `int` и размер этого массива типа `int`) — возвращает сумму модулей элементов массива, расположенных от первого чётного элемента и до последнего нечётного, включая первый и не включая последний. .
- `sum_before_even_and_after_odd` (на вход массив типа `int` и размер этого массива типа `int`) — возвращает сумму модулей элементов массива, расположенных до первого чётного элемента (не включая элемент) и после последнего нечётного (включая элемент).

Ввод значений происходит в цикле `while`, сначала вводится значение `n` затем заполняется `list`.

При помощи оператора `switch`, в зависимости от значения переменной `n`, функцией `printf` выводится на консоль:

- 0 : индекс первого чётного элемента. (`index_first_even`)
- 1 : индекс последнего нечётного элемента. (`index_last_odd`)
- 2 : сумма модулей элементов массива, расположенных от первого чётного элемента и до последнего нечётного, включая первый и не включая последний. (`sum_between_even_odd`)

- 3 : сумма модулей элементов массива, расположенных до первого чётного элемента (не включая элемент) и после последнего нечётного (включая элемент). (`sum_before_even_and_after_odd`)
- Иначе выводится строка "Данные некорректны".

Разработанный программный код см. в приложении А.

### **Выводы.**

Были изучены условия, циклы, оператор `switch`, работа с функциями в языке программирования С.

Разработана программа, выполняющая считывание исходных с помощью функции `scanf()` и цикла `while(){}` в переменную `n` и массив `list[100]`, условием которого было равенство переменной `s`, хранящей код символа между числами, коду символа пробела, написаны функции для обработки входных результатов, подробное описание которых приведено в разделе «выполнение работы», с помощью оператора `switch(){}` и функции `printf()` реализован вывод результата определённой функции в зависимости от входного управляющего значения `option`:

- если `n = 0` — выводится результат функции `int index_first_even();`
- если `n = 1` — выводится результат функции `int index_last_odd();`
- если `n = 2` — выводится результат функции `int sum_between_even_odd();`
- если `n = 3` — выводится результат функции `int sum_before_even_and_after_odd();`

Если значение `n` не соответствует ни одному из перечисленных — выводится строка «Данные некорректны».

## ПРИЛОЖЕНИЕ А

### ИСХОДНЫЙ КОД ПРОГРАММЫ

Название файла: 1.c

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

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

int index_last_odd(int list[], int listsize){
    for (int i = listsize - 1; i >= 0; i--){
        if(abs(list[i])%2==1) {
            return i;
        }
    }
}

int sum_between_even_odd(int list[], int listsize){
    int res = 0;

    for (int i = index_first_even(list, listsize); i <
index_last_odd(list, listsize); i++){
        res += abs(list[i]);
    }

    return res;
}

int sum_before_even_and_after_odd(int list[], int listsize){
    int sum = 0;

    for (int i = 0; i < listsize; i++) sum+= abs(list[i]);

    return sum - sum_between_even_odd(list, listsize);
}

int main(){
    int n = -1, list[100]; // n - значение для функции, list -
список для вводимых символов
    int listsize = 0; // насколько массив заполнен

    char sym = ' ';
    while(listsize<100 && sym == ' '){ //Ввод
        if (n == -1) scanf("%d%c", &n, &sym);
        else scanf("%d%c", &list[listsize++], &sym);
    }
}
```

```

switch (n)
{
case 0:
    printf("%d\n", index_first_even(list, listsize));
    break;
case 1:
    printf("%d\n", index_last_odd(list, listsize));
    break;
case 2:
    printf("%d\n", sum_between_even_odd(list, listsize));
    break;
case 3:
    printf("%d\n", sum_before_even_and_after_odd(list,
listsize));
    break;
default:
    printf("Данные некорректны\n");
    break;
}
return 0;
}

```