

**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. 1304

Радионов Б. С.

---

Преподаватель

Чайка К. В.

---

Санкт-Петербург

2021

### **Цель работы.**

Исследование управляющих конструкций языка Си и изучение основных способов написания программы.

### **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 нулевого элемента. (index\_first\_zero)
- 1 : индекс последнего нулевого элемента. (index\_last\_zero)
- 2 : Найти сумму модулей элементов массива, расположенных от первого нулевого элемента и до последнего. (sum\_between)
- 3 : Найти сумму модулей элементов массива, расположенных до первого нулевого элемента и после последнего. (sum\_before\_and\_after)

иначе необходимо вывести строку "Данные некорректны".

### **Выполнение работы.**

Программа считывает значение, введенное пользователем с клавиатуры, и должна выдать ответ, в зависимости от введенного значения. Допустимые значения – 0, 1, 2, 3. При иных значениях, программа выводит надпись “Данные некорректны”.

- При введенном значении “0”, программа вычисляет индекс первого нулевого элемента в массиве.
- При введенном значении “1”, программа вычисляет индекс последнего нулевого элемента в массиве.
- При введенном значении “2”, программа вычисляет сумму элементов между первым и последним нулевым элементом массива (не включительно).

- При введённом значении “3”, программа вычисляет сумму элементов до первого нулевого элемента и после последнего нулевого элемента массива (не включительно).

Переменные, используемые в коде, для решения поставленных задач:

1. *value* – переменная, которой присваивается введенное значение пользователем.
2. *var\_result* – переменная, принимающая результат функций.
3. *arr[N]* – массив, размером не больше *N*.
4. *n* – переменная, принимающая значение размера введённого массива.
5. *i* – переменная, являющаяся счетчиком в цикле функции по определению размера введённого массива.
6. *c* – переменная, отслеживающая символ перевода строки.
7. *func\_value* – переменная, возвращающая значения функций.
8. *d* – переменная, используемая в отдельных функциях, принимающая значение функции по определению индекса первого нуля.
9. *b* – переменная, используемая в отдельных функциях, принимающая значение функции по определению индекса последнего нуля.
10. *sum* – переменная, хранящая в себе сумму элементов между первым и последним нулём.
11. *sum1* – переменная, хранящая в себе сумму элементов до первого нуля.
12. *sum2* – переменная, хранящая в себе сумму элементов после первого нуля.

Функции, используемые в коде, для решения поставленных задач:

1. *main* – главная функция, считывающая значение переменной *value*, в зависимости от которой будет определяться с помощью условного оператора *switch*, какую из четырёх функций нужно

выполнить и какой ответ вывести на экран. Если введено значение, отличное от 0, 1, 2, 3 – функция выводит на экран надпись “Данные некорректны”.

2. *size\_of\_massive* – функция для расчёта размера массива. В ней с помощью цикла *for(i = 0; i < n; i++)* происходит считывание массива по индексам его элементов до символа перевода строки, если встречается символ перевода строки – цикл заканчивается. После этого функция возвращает нам размер массива посредством прибавления единицы к *i*, так как считывание массива происходит до предпоследнего элемента.
3. *index\_first\_zero* – функция для определения индекса первого нулевого элемента массива. В ней с помощью цикла *for(i = 0; i < n; i++)* идет проверка на элемент со значением нуль. Если встречается такой элемент, цикл досрочно прерывается с помощью *break*. Функция возвращает значение такого индекса в переменную *func\_value*.
4. *index\_last\_zero* – функция для определения индекса последнего нулевого элемента массива. В ней так же используется цикл *for(i = 0; i < n; i++)*, если встречается элемент со значением нуль – *func\_value* принимает значение его индекса, и такая проверка идёт до последнего элемента массива.
5. *sum\_between* – функция, которая определяет сумму элементов массива между первым и последним нулём. В переменные *d* и *b* передаются значения функций *index\_first\_zero* и *index\_last\_zero*. Переменной *sum* присваивается значение 0. В цикле *for* переменной *i* присваивается значение переменной *d*. С каждой итерацией *i* увеличивается на 1, а также переменная *sum* увеличивается на значение текущего элемента. Цикл выполняется ровно до тех пор, пока *i* не примет значение переменной *b*.

Функция возвращает значение суммы элементов между первым и последним нулём в переменной *func\_value*.

6. *sum\_before\_and\_after* – функция для определения суммы элементов до первого нуля и после последнего нуля. В данной функции переменным *sum1* и *sum2* присваивается значение 0. В переменные *d* и *b* передаются значения функций *index\_first\_zero* и *index\_last\_zero*. Здесь используются два цикла. Первый цикл *for(int i = 0; i < d; i++)* рассчитывает сумму элементов до первого нуля и сохраняет значение суммы в переменную *sum1*. Второй цикл *for(int i = b; i < n; i++)* рассчитывает сумму элементов после последнего нуля и сохраняет значение суммы в переменную *sum2*. Далее в переменную *func\_value* сохраняется значение суммы переменных *sum1* и *sum2*.

### Тестирование.

Результаты тестирования представлены в табл. 1.

Таблица 1 – Результаты тестирования

№ п/п	Входные данные	Выходные данные	Комментарии
1.	0 1 2 0 3 4 0 5	2	Выполнен успешно
2.	1 2 3 0 2 0 9 0 8	6	Выполнен успешно
3.	2 1 0 2 -2 1 0 1	5	Выполнен успешно
4.	3 -4 2 3 0 1 2 0 2 -7 1	19	Выполнен успешно
5.	4 1 0 9 0 2 3	Данные некорректны	Выполнен успешно

### Выводы.

В ходе лабораторной работы мы исследовали управляющие конструкции языка Си и изучили основные конструкции путём написания программы.

Разработана программа, выполняющая считывание исходных данных с клавиатуры (массив из целых чисел и значение). Для обработки команд пользователя использовался условный оператор *switch*.



