

**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

Мусаев А. И.

---

Преподаватель

Чайка К. В.

---

Санкт-Петербург

2021

## **Цель работы.**

Исследование управляющих конструкций, изучение основных способов написания программы и начало изучения функционального программирования.

## **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 массиве есть хотя бы один четный и нечетный элемент.

В зависимости от значения, функция должна выводить следующее:

- 0: индекс первого четного элемента. (index\_first\_even)
- 1: индекс последнего нечетного элемента. (index\_last\_odd)
- 2: Найти сумму модулей элементов массива, расположенных от первого четного элемента и до последнего нечетного, включая первый и не включая последний. (sum\_between\_even\_odd)
- 3: Найти сумму модулей элементов массива, расположенных до первого четного элемента (не включая элемент) и после последнего нечетного (включая элемент). (sum\_before\_even\_and\_after\_odd)
- Иначе необходимо вывести строку "Данные некорректны".

## **Выполнение работы.**

Задачи:

В зависимости от введенного пользователем значения программа должна выводить разные ответы:

1. Если пользователь ввёл 0, программа должна вывести индекс первого четного элемента массива.
2. Если пользователь ввёл 1, программа должна вывести индекс последнего нечетного элемента массива.
3. Если пользователь ввёл 2, программа должна найти сумму модулей элементов массива, расположенных между первым четным и последним нечетным, включая первый четный и не включая последний нечетный.

4. Если пользователь ввёл 3, программа должна найти сумму модулей элементов массива, расположенных до первого четного, не включая его, и после последнего нечетного, включая его.

Используемые переменные:

Целочисленные переменные:

- 1) *i* — переменная счетчик
- 2) *task\_value* — введенное пользователем значение для постановки задачи

задачи

- 3) *array\_length* — количество элементов, поданных в массив
- 4) *index\_first\_even* — номер первого четного элемента
- 5) *index\_last\_odd* — номер последнего нечетного элемента
- 6) *sum\_between\_even\_odd* — сумма модулем элементов массива между

первым четным и последним нечетным(включая первый, не включая последний)

7) *sum\_before\_even\_and\_after\_odd* — сумма модулей элементов массива до 1 четного и после последнего нечетного(не включая первый, включая последний)

Символьные переменные:

*sep* — переменная для считывания пробелов и символов переноса строки между элементами массива.

Функции:

*main* — основная функция. Она считывает значение *task-value*, потом считывает массив. Вызывает две функции *find\_index\_first\_even* и *find\_index\_last\_odd*, которые я опишу ниже, и записывает их значения в переменные *index\_first\_even* и *index\_last\_odd* соответственно. Следом в зависимости от значения *task\_value* она выводит разные значения:

- 1) *index\_first\_even* («Данные некорректны» в случае, если массив некорректен), если *task\_value* равен 0.
- 2) *index\_last\_odd* («Данные некорректны» в случае, если массив некорректен), если *task\_value* равен 1.

- 3) Вызывает функцию *output\_sum\_between\_even\_odd*, если *task\_value* равно 2.
- 4) Вызывает функцию *output\_sum\_before\_even\_and\_after\_odd*, если *task\_value* равно 3.

*find\_index\_first\_even* — функция, возвращающая первый четный элемент в массиве или -1, если такового нет. Запускается цикл, который проходит по всему массиву, как только он находит четный элемент, записывает его *index\_first\_even* и выходит из цикла. Функция возвращает значение *index\_first\_even*. Не выводит она его, так как это значение может понадобиться в других функциях и его вывод логичнее оформить в *main*, чтобы не писать две одинаковые функции.

На вход принимает сам массив и его длину.

*find\_last\_odd* — функция, возвращающая последний нечетный элемент в массиве или -1, если такового нет. Запускается цикл, который проходит по всему массиву, во время цикла проверяется условие, что элемент нечетен, если он нечетен, элемент массива записывает в переменную *index\_last\_odd*. Функция возвращает значение из *index\_last\_odd*. Не выводит он его по той же причине, что и *find\_index\_first\_even*.

На вход принимает сам массив и его длину.

*output\_sum\_between\_even\_odd* – функция, выводящая в окно вывода сумму между первым четным и последним нечетным элементом, включая первый и не включая последний.

Она принимает на вход сам массив, номера первого четного и последнего нечетного элемента. Сначала она проверяет, есть ли вообще такие элементы в массиве, если нет, выводит сообщение о том, что данные некорректны, иначе с помощью цикла считает сумму модулей элементов массива, стоящих на нужных местах.

*output\_sum\_before\_even\_and\_after\_odd* – функция, выводящая в окно вывода сумму до первого четного и после последнего нечетного, исключая первый, включая последний.

Она принимает на вход сам массив, номера первого четного и последнего нечетного элемента. Сначала она проверяет, есть ли в массиве четный и нечетный элемент, если нет, то выводит сообщение о том, что данные некорректны, иначе с помощью цикла сначала считает сумма от нулевого до первого четного-1, потом с помощью второго цикла от последнего нечетного включительно и до последнего элемента массива, всё это прибавляет к той же сумме. Полученную сумму выводит на экран.

### **Тестирование.**

Результаты тестирования представлены в табл. 1.

Таблица 1 – Результаты тестирования

№ п/п	Входные данные	Выходные данные	Комментарии
1.	0 -8 -23 -30 -11 -28 15 -20 -24 -27 5 -13 5 21 -5 16 30 -12 15 -14 -28 -27 -11 -5 4 29 -5	0	Правильно номер первого четного элемента – это 0
2.	1 -8 -23 -30 -11 -28	4	Правильно, номер последнего нечетного элемента – это 4
3.	2 -8 -23 -30 -11 -28	61	Правильно, сумма модулей от -8 до -11 равна 61
4.	3 -8 -23 -30 -11 -28	39	Правильно, сумма модулей от до -8 и после -11 равна 39

### **Выводы.**

Я исследовал управляющие конструкции языка Си, изучил основные способы написания программ и начал изучения функционального программирования.