**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. 3342 |  | Романов Е.А. |
| Преподаватель |  | Глазунов С.А. |

Санкт-Петербург

2023

## Цель работы

Изучение возможностей практического применения языка программирования C, с использованием таких языковых конструкций, как циклы, условия, а также оператора switch.

## 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 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".  
Ошибкой в данном задании считается дублирование кода!

Подсказка: функция нахождения модуля числа находится в заголовочном файле stdlib.h стандартной библиотеки языка Си.

При выводе результата, не забудьте символ переноса строки.

## Выполнение работы

Массив целых чисел подаётся в виде строки. С помощью функции fgets и цикла for числа, образованные последовательностью символов, преобразуются в целые, однако они считываются справа на лево, чтобы это исправить используется функция reverseNum, получившиеся значения записываются в массив nums.

В функции main с помощью оператора switch устанавливается какая операция с массивом должна быть выполнена. В случае, если на вход номер операции, не соответствующий диапазону от 0 до 3, то выводится сообщение "Данные некорректны". Вывод данных осуществляется с помощью функции printf.

Функция index\_first\_even принимает массив и его длину, с помощью цикла for элементы переданного массива перебираются до тех пор, пока не попадётся чётное значение. Индекс найденного значения возвращается и выводится на экран.

Функция index\_last\_odd принимает массив и его длину, с помощью цикла for производится перебор элементов массива, индексы найденных нечётных элементов записываются в локальную переменную lastInd, последний записанный результат будет являться искомым, он возвращается функцией и выводится на экран.

Функция sum\_between\_even\_odd принимает массив и его длину, с помощью цикла for производится перебор массива, начиная с первого чётного элемента, включая его самого, для поиска индекса которого применяется функция index\_first\_even, и заканчивая последним нечётным элементом, который не входит в искомую сумму, его индекс находится с помощью функции index\_last\_odd. Модули попавших в этот промежуток элементов массива суммируются и в локальной переменной sum, значение которой и возвращает функция.

Функция sum\_before\_even\_and\_after\_odd принимает массив и его длину, с помощью цикла for производится перебор и последовательное суммирование модулей всех его элементов, после чего из найденной суммы вычитается результат работы функции sum\_between\_even\_odd. Найденное значение и является искомым, его и возвращает функция.

Переменные, используемые в программе:

-nums массив целых чисел, заполняемый на основе вводимых данных

-string массив символов, в который помещается строка, содержащая целые числа

-size константа, отражающая размер массива string

-powIndex отражает количество разрядов передаваемых чисел

-newnum содержит числа, преобразованные из строк в целочисленный тип данных

-negativeNum флаг, содержащий информацию о том, является ли конкретное число отрицательным

-zeroendingnum содержит информацию о том, оканчивается ли число на 0

-newnumindex используется для заполнения массива nums и для хранения его длины

-programMode содержит информацию о том, какая операция с массивом должна быть выполнена

Функции, используемые в программе:

- int reverseNum(int num, int negativeNum, int zeroendingnum) возвращает перевёрнутое слева на право значение числа

- int index\_first\_even(int arr[100], int l) возвращает индекс первого чётного числа

- int index\_last\_odd(int arr[100], int l) возвращает индекс последнего нечётного числа

- int sum\_between\_even\_odd(int arr[100], int l) возвращает сумму элементов между первым чётным, включая его значение, и последним нечётным, не включая его значение

- int sum\_before\_even\_and\_after\_odd(int arr[100], int l) возвращает сумму элементов до первого чётного, не включая его значение, и от первого нечётного, включая его значение.

## Тестирование

Результаты тестирования представлены в табл. 1.

Таблица 1 – Результаты тестирования

|  |  |  |
| --- | --- | --- |
| № п/п | Входные данные | Выходные данные |
|  | 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 |
|  | 1 29 4 -5 -11 -27 -28 15 -8 -23 -30 -11 -28 15 -20 -24 -27 5 -13 5 21 -5 16 30 -12 | 20 |
|  | 2 -12 30 16 -5 -13 -27 29 4 -5 -11 -27 -28 15 -8 -23 -30 -11 -28 15 -20 -24 | 322 |
| 4. | 3 30 -23 8 4 74 -87 16 -5 -13 -27 29 4 -5 -11 -27 -28 15 -8 -23 -30 -11 -28 15 -20 -24 58 10 -10 3 | 3 |

## Выводы

Были изучены основные алгоритмические и управляющие конструкции языка C: условные операторы if и switch, циклы for и while.

Была разработана программа, выполняющая считывание с клавиатуры данных и обрабатывающая их заданным образом. Данные считываются с помощью функции fgets и обрабатываются циклами for. Для определения типа выполняемых операций используется оператор switch. Операции определены в отдельные функции, в которых с помощью циклов for и условных операторов происходит обработка полученных от пользователя данных. Результат работы функций выводится на экран.

# **Приложение А Исходный код программы**

Название файла: main.c

#include <stdlib.h>

#include <stdio.h>

int reverseNum(int num, int negativeNum, int zeroendingnum) {

float num\_clone = num;

int newnum = 0;

int digitscounter = 0;

while (num\_clone >= 1) {

num\_clone /= 10;

digitscounter++;

}

for (int i = 0; i < digitscounter; i++) {

int p = num % 10;

int powerIndex = 1;

num /= 10;

for (int j = 1; j < digitscounter-i; j++) {

powerIndex \*= 10;

}

newnum += p \* powerIndex;

}

for (int g = 0; g < zeroendingnum; g++) {

newnum \*= 10;

}

zeroendingnum = -1;

if (negativeNum) {

return newnum \*(- 1);

}

return newnum;

}

int index\_first\_even(int arr[100], int l) {

for (int i = 0; i < l; i++) {

if (arr[i] % 2 == 0) {

return i;

}

}

}

int index\_last\_odd(int arr[100], int l) {

int lastInd = 0;

for (int i = 0; i < l; i++) {

if (abs(arr[i]) % 2 == 1) {

lastInd = i;

}

}

return lastInd;

}

int sum\_between\_even\_odd(int arr[100], int l) {

int sum = 0;

for (int i = index\_first\_even(arr, l); i < index\_last\_odd(arr, l); i++) {

sum += abs(arr[i]);

}

return sum;

}

int sum\_before\_even\_and\_after\_odd(int arr[100], int l) {

int sum = 0;

for (int i = 0; i < l; i++) {

sum += abs(arr[i]);

}

return sum - sum\_between\_even\_odd(arr, l);

}

int main() {

int nums[100];

const int size = 1000;

char string[size];

fgets(string, 1000, stdin);

int powIndex = 1, newnum = 0, negativeNum = 0, zeroendingnum=-1,newnumindex=0;

int programMode = string[0] - '0';

for (int i = 2; i < size; i++) {

if ((string[i] != ' ') && (string[i] != '\n') && (string[i]!='-')) {

newnum += (string[i] - '0') \* powIndex;

powIndex \*= 10;

if (string[i] == '0') {

zeroendingnum += 1;

}

else {

zeroendingnum = 0;

}

}

else if (string[i] == '-') {

negativeNum = 1;

}

else if (string[i] == '\n') {

nums[newnumindex] = reverseNum(newnum, negativeNum, zeroendingnum);

newnumindex++;

break;

}

else {

nums[newnumindex] = reverseNum(newnum, negativeNum, zeroendingnum);

newnumindex++;

if (negativeNum) {

negativeNum = 0;

}

powIndex = 1;

newnum = 0;

}

}

switch (programMode) {

case(0) :

printf("%d",index\_first\_even(nums, newnumindex));

break;

case(1) :

printf("%d", index\_last\_odd(nums, newnumindex));

break;

case(2) :

printf("%d", sum\_between\_even\_odd(nums, newnumindex));

break;

case(3) :

printf("%d", sum\_before\_even\_and\_after\_odd(nums, newnumindex));

break;

default :

printf("%s", "Данные некорректны");

}

return 0;

}