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 _____ Заика Т.П.

Преподаватель _____ Чайка К.В.

Санкт-Петербург

2021

Цель работы.

Научиться применять условия, циклы и оператор switch

Задание.

Вариант 3.

Напишите программу, выделив каждую подзадачу в отдельную функци

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".
```

Основные теоретические положения.

Условия

Циклы

Оператор switch

Выполнение работы.

В ходе решения задачи были определены переменные для входных данных, осуществлен ввод с клавиатуры для определения значения переменной, отвечающей за режим работы программы, а также для определения значений внутри массива. После получения входных данных, создано условие для проверки корректного значения переменной режима программы. Далеее в зависимости от значения данной переменной, идет

обработка массива. Для обработки массива было решено сделать четыре отдельных функции, решающие поставленные задачи. После обработки массива одной из функций, происходит вывод результата на экран, и программа завершается.

Переменные:

task_number — определение режима работы программы

arr_size — размер массива

arr[] - массив целочисленных значений

stop_char — перменная для отслеживания окончания ввода строки, проверяется на наличие в себе символа перевода строки

iter_num — используется для подсчета кол-ва элементов в массиве, которые были занесены через ввод с клавиатуры

task_answer — хранит ответ на текущее задание

answer_index_first_zero — хранит ответ на задание 0, используется для возвращения при вызове функции

answer_index_last_zero - хранит ответ на задание 1, используется для возвращения при вызове функции

iter — итерируемая переменная

answer_sum_between - хранит ответ на задание 2, используется для возвращения при вызове функции

answer_sum_before_after - хранит ответ на задание 3, используется для возвращения при вызове функции

index_start_zero — хранит значение индекса первого нуля в массиве

index_last_zero — хранит значение индекса последнего нуля в массиве

module_num — хранит значение модуля текущего элемента массива

Функции:

Все функции в качестве аргументов принимают arr[] - массив элементов, и arr_size — размер массива (кол-во значащих элементов массива)

index_first_zero — находит индекс первого нулевого элемента.

index_last_zero — находит индекс последнего нулевого элемента.

sum_between — находит сумму модулей элементов массива, расположенных от первого нулевого элемента и до последнего.

sum_before_and_after — находит сумму модулей элементов массива, расположенных до первого нулевого элемента и после последнего.

Разработанный программный код см. в приложении А.

Тестирование.

Результаты тестирования представлены в табл. 1.

Таблица 1 – Результаты тестирования

№ п/п	Входные данные	Выходные данные	Комментарии
1.	0 -21 10 0 -23 -7 -15 -14 8 -9 10 -13 -14 -27 0 -7 12 -18	2	Успешный тест
2.	1 -21 10 0 -23 -7 -15 -14 8 -9 10 -13 -14 -27 0 -7 12 -18	13	Успешный тест
3.	2 -21 10 0 -23 -7 -15 -14 8 -9 10 -13 -14 -27 0 -7 12 -18	140	Успешный тест
4.	3 -21 10 0 -23 -7 -15 -14 8 -9 10 -13 -14 -27 0 -7 12 -18	68	Успешный тест
5.	4 -21 10 0 -23 -7 -15 -14 8 -9 10 -13 -14 -27 0 -7 12 -18	Данные некорректны	Успешный тест

Выводы.

Были исследованы, изучены условия, циклы, оператор switch

Разработана программа, выполняющая считывание с клавиатуры исходных данных и команды пользователя. После считывания первого значения, номер задания, происходит считывание строки, которая обрабатывается функцией в зависимости от номера задания, например 0 — получение индекса первого нулевого элемента массива, 1 — получение индекса последнего нулевого массива, 2 — нахождение суммы модулей элементов массива, расположенных от первого нулевого элемента и до последнего, 3 — нахождение суммы модулей элементов массива, расположенных до первого нулевого элемента и после последнего. В конце программа должна выдать корректный ответ на входные данные.

Задача была решена при помощи условий if-else, применяемых для обработки входных данных, введенных пользователем, циклов for и while, применяемых для работы с элементами массива, а также опретора switch, определяющего режим работы программы.

ПРИЛОЖЕНИЕ А

ИСХОДНЫЙ КОД ПРОГРАММЫ

Название файла: main.c

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


int index_first_zero(int arr[], int arr_size){
      int answer_index_first_zero = 0;

      for (int i=0; i<arr_size; i++){
            if (arr[i] == 0){
                  answer_index_first_zero = i;
                  break;
            }
      }

      return answer_index_first_zero;
}

int index_last_zero(int arr[], int arr_size){
      int answer_index_last_zero = 0;
      int iter = 0;

      while(iter<arr_size){
            if (arr[iter] == 0){
                  answer_index_last_zero = iter;
            }
            iter++;
      }

      return answer_index_last_zero;

}

int sum_between(int arr[], int arr_size){
      int answer_sum_between = 0;

      int index_start_zero = index_first_zero(arr, arr_size);
      int index_finish_zero = index_last_zero(arr, arr_size);

      int module_num = 0;

      for (int i = index_start_zero; i<index_finish_zero; i++) {
            module_num = abs(arr[i]);
            answer_sum_between += module_num;
      }

      return answer_sum_between;
}

int sum_before_after(int arr[], int arr_size){
      int answer_sum_before_after = 0;
```

```
	int iter = 0;

	int index_start_zero = index_first_zero(arr, arr_size);
	int index_finish_zero = index_last_zero(arr, arr_size);

	int module_num = 0;

	while (iter<index_start_zero){
		module_num = abs(arr[iter]);
		answer_sum_before_after += module_num;
		iter++;
	}

	while(index_finish_zero<arr_size){
		module_num = abs(arr[index_finish_zero]);
		answer_sum_before_after += module_num;
		index_finish_zero++;
	}

	return answer_sum_before_after;
}


int main(){
	int task_number = 0;

	scanf("%d", &task_number);

	if (task_number < 0 || task_number > 3) {
		printf("Данные некорректны\n");
	} else {
		int arr_size = 100;
		int arr[arr_size];
		char stop_char;

		int iter_num = 0;

		while (iter_num < arr_size) {
			scanf("%d%c", &arr[iter_num], &stop_char);
			iter_num++;
			if (stop_char == '\n'){
				break;
			}
		}

		arr_size = iter_num;

		int task_answer = 0;

		switch (task_number) {
			case 0:
				task_answer=index_first_zero(arr, arr_size);
				printf("%d\n", task_answer);
				break;
			case 1:
				task_answer=index_last_zero(arr, arr_size);
```

```
                    printf("%d\n", task_answer);
                    break;
               case 2:
                    task_answer=sum_between(arr, arr_size);
                    printf("%d\n", task_answer);
                    break;
               case 3:
                    task_answer=sum_before_after(arr, arr_size);
                    printf("%d\n", task_answer);
                    break;
          }
     }

     return 0;
}
```