

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. 0304

Маркуш Н.В.

Преподаватель

Чайка К.В.

Санкт-Петербург

2021

Цель работы.

Изучить основные управляющие конструкции языка C (условия, циклы, оператор switch), массивы и функции.

Задание.

Вариант №3. 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".

Основные теоретические положения.

Использованы управляющие конструкции языка C: *for*, *if-else*, *while*, *switch*.
Использованы заголовочные файлы `<stdio.h>` и `<stdlib.h>`. Из `<stdio.h>` были использованы функции *scanf()*, *getchar()*. Из `<stdlib.h>` был использован *abs()* для возвращения модуля переменной.

Выполнение работы.

1. Подключение заголовочных файлов `<stdio.h>` и `<stdlib.h>` при помощи директивы препроцессора *#include*.

2. Задаётся макроопределение *MAX_SIZE* с помощью директивы препроцессора *#define*.

3. Объявляются 4 функции, для дальнейшей их проверки в управляющей конструкции *switch*. Они получают в качестве аргументов массив из целых чисел *values[]*, целочисленную переменную *size*, определяющую длину массива, а также переменную *act*, которая определяет выполняемое действие

4. Функция *int main()*.

a. Присваивается значение *int act* с помощью функции *scanf()*.

b. Создаётся переменная *int act* (для хранения значения 0,1,2,3), переменная *int i* (счётчик для количества чисел в массиве *input[]*), *int size* (количество введённых чисел) и массив *int input[MAX_LENGTH]*.

c. В цикле *for* происходит присваивание считываемых с клавиатуры значений каждому элементу массива с индексом *int i*.

Цикл продолжит принимать значения до тех пор, пока не будет встречен символ конца строки '\n' или не будет введено максимальное число элементов *MAX_LENGTH*, переменная *int i* увеличивается на 1 за каждый тур цикла.

d. Далее объявляем массив *valuse[size]*. В ещё одном цикле *for* происходит перенос значений из массива *input[]* в массив *values[]*.

5. Управляющая конструкция *switch*. В ней проверяется значение *int x*, в соответствии с которой будет выполняться одна из четырёх функций. Если *int x* будет принимать значения *int x < 0* или *int x > 3*, то на экран выводится сообщение «Данные некорректны».

6. Функция *index_first_zero(int values[], int act, int size)*. В функции при помощи цикла *for* перебираются элементы, при нахождении первого нулевого элемента, функция возвращает индекс данного элемента и выводит его на экран.

7. Функция *index_last_zero(int values[], int act, int size)*. В функции при помощи цикла *for* перебираются элементы с тем лишь отличием, что перебираются они с конца. При нахождении последнего(для списка)/первого(для

цикла) нулевого элемента, функция возвращает индекс данного элемента и выводит его на экран.

8. Функция *sum_between(int values[], int act, int size)*.

а. В функции объявляются переменные *int first* и *int last* в которые присваиваются значения функций *index_first_zero(values, act, size)* и *index_last_zero(values, act, size)* соответственно (для нахождения индекса первого и последнего элемента). Объявляется переменная *int sum* для нахождения суммы.

б. Цикл *for*. В цикле перебираются все значения массива от элемента с индексом *int first_even* до элемента с индексом *int last_odd* не включая последний. Модуль значения каждого элемента прибавляется к переменной *int sum*.

с. В конце функция возвращает и выводит на экран *int sum*.

9. Функция *sum_before_and_after(int values[], int act, int size)*.

а. В функции объявляются переменные *int all_sum* (сумма всех элементов массива). Объявляется переменная *int sum* для нахождения суммы.

б. Первый цикл *for*. В цикле перебираются все значения массива от нулевого до последнего. Модуль значения каждого элемента прибавляется к переменной *int all_sum*.

с. Затем присваиваем переменной *int sum* значение выражения (*all_sum - sum_between(int values[], int act, int size)*)

д. В конце функция выводит на экран *int sum*.

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 0 -23 -30 -11 -28 15	0	

	-20 -24 -27 5 -13 5 21 -5		
2.	1 23 3 4 5 0 32 42 5 23 0 23 0 2 0	13	
3.	2 13 0 2 39 1 82 01 9 2 92 83 0 18 -2 3 0 -1 3 0	338	
4.	3 -2 3 93 9 0 219 0 18 -2 93 18 0 18 2 01 82 1 -29	240	

ПРИЛОЖЕНИЕ А ИСХОДНЫЙ КОД ПРОГРАММЫ

Название файла: lab.1.1.c

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

#define MAX_SIZE 100

int index_first_zero(int values[], int act, int size){
    int truth = 0;
    int first = 0;

    for(int i = 0; i < size; i++){
        if(values[i] == 0){
            first = i;
            if (act == 0){
                printf("%d\n", first);
            }
            truth = 1;
            break;
        }
    }
    if(truth == 0){
        printf("Данные некорректны\n");
    }
    return first;
}

int index_last_zero(int values[], int act, int size){
    int truth = 0;
    int last = 0;

    for(int i = (size - 1); i >= 0; i--){
        if(values[i] == 0){
            last = i;
            if(act == 1){
```

```

        printf("%d\n", last);
    }
    truth = 1;
    break;
}
}
if(truth == 0){
    printf("Данные некорректны\n");
}
return last;
}

int sum_between(int values[], int act, int size){
    int first = index_first_zero(values, act, size);
    int last = index_last_zero(values, act, size);
    int sum = 0;

    for(int i = first; i < last; i++){
        sum += abs(values[i]);
    }
    if(act == 2){
        printf("%d\n", sum);
    }
    return sum;
}

void sum_before_and_after(int values[], int act, int size){
    int all_sum = 0;
    for(int i = 0; i < size; i++){
        all_sum += abs(values[i]);
    }
    int sum;
    sum = all_sum - sum_between(values, act, size);
    printf("%d\n", sum);
}

```

```

int main(){
    int act;
    scanf("%d", &act);

    char space;
    int input[MAX_SIZE];
    int size = 0;
    for(int i = 0; i < MAX_SIZE; i++){
        scanf("%d%c", &input[i], &space);
        size++;
        if(space == '\n'){
            break;
        }
    }
    int values[size];
    for(int i = 0; i < size; i++){
        values[i] = input[i];
    }

    switch ( act ){
        case 0:
            index_first_zero(values, act, size);
            break;
        case 1:
            index_last_zero(values, act, size);
            break;
        case 2:
            sum_between(values, act, size);
            break;
        case 3:
            sum_before_and_after(values, act, size);
            break;
        default:
            printf("Данные некорректны\n");
    }
}

```



```
    return 0;  
}
```