**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

|  |  |  |
| --- | --- | --- |
| Студентка гр. 9304 |  | Каменская Е.К. |
| Преподаватель |  | Чайка К.В. |

Санкт-Петербург

2019

## Цель работы.

Создать собственные функции, изучить работу условий, циклов и операторов языка C и применить их в собственной программе.

## 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 : максимальное по модулю число в массиве. (abs\_max)

1 : минимальное по модулю число в массиве. (abs\_min)

2 : разницу между максимальным по модулю и минимальным по модулю элементом. (diff)

3 : сумму элементов массива, расположенных после максимального по модулю элемента (включая этот элемент). (sum)

иначе необходимо вывести строку "Данные некорректны".

## Основные теоретические положения.

Раздел выполняется в соответствии с указаниями преподавателя.

В разделе может быть приведено описание используемых управляющих конструкций языка, используемых встроенных (библиотечных) структур данных, описание существующих алгоритмов, которые будут реализованы в работы со ссылкой на источник.

Этот раздел может отсутствовать.

## Выполнение работы.

Переменные:

* *arr[]* – массив вводимых целых чисел. Начальный размер *arr[]* равен 100 (максимальное возможное количество элементов по условию).
* *real\_n* – целочисленная переменная для хранения действительного количества введенных элементов.
* *option* – целочисленная переменная для хранения значения, от которого зависит вывод программы.
* *c* – символьная переменная, считывающая символ, следующий за введенным.

Функции:

* *int abs\_max(int \*arr, int real\_n)* принимает на вход указатель на первый элемент массива *arr[]* и целочисленное количество элементов в нем. Внутри функции создается целочисленная переменная *a\_max*, инициализируемая как модуль первого элемента массива *arr[]*, для записи максимального модуля и целочисленная переменная *elem* для записи искомого элемента. С помощью цикла *for* и оператора *if* функция сравнивает модуль каждого элемента с *a\_max* и присваивает *elem* значение элемента с максимальным модулем. В результате возвращается значение *elem*.
* *int abs\_min(int \*arr, int real\_n)* принимает на вход указатель на первый элемент массива *arr[]* и целочисленное количество элементов в нем. Внутри функции создается целочисленная переменная *a\_min*, инициализируемая как модуль первого элемента массива *arr[]*, для записи наименьшего модуля и целочисленная переменная *elem* для записи искомого элемента. С помощью цикла *for* и оператора *if* функция сравнивает модуль каждого элемента с *a\_min* и присваивает *elem* значение элемента с минимальным модулем. В результате возвращается значение *elem*.
* *int diff(int \*arr, int real\_n)* принимает на вход указатель на первый элемент массива *arr[]* и целочисленное количество элементов в нем. Внутри функции создается целочисленная переменная *e\_max*, инициализируемая как результат работы функции *abs\_max*, для записи числа с максимальным модулем и целочисленная переменная *e\_min*, инициализируемая как результат работы функции *abs\_min*, для записи числа с минимальным модулем. На выходе возвращается значение разности *e\_max* и *e\_min*.
* *int sum(int \*arr, int real\_n)* принимает на вход указатель на первый элемент массива *arr[]* и целочисленное количество элементов в нем. Внутри функции создается целочисленная переменная *e\_max*, инициализируемая как результат работы функции *abs\_max*, для записи числа с максимальным модулем и целочисленная переменная *answer*, инициализируемая нулем, для записи суммы. С помощью цикла *for* и оператора *if* начиная с элемента, равного *e\_max*, все элементы прибавляются к *answer*:

for(int i=0; i<real\_n; i++){

if((arr[i]==e\_max)||(e\_max==0)){

answer += arr[i];

e\_max = 0;

}

}

В итоге возвращается переменная *answer*.

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 |
|  | 2 -90 16 77 2 34 | -92 | Найдена разница между максимальным по модулю и минимальным по модулю элементом. |
|  | 3 0 0 -2 0 | -2 | Сумма всех элементов, начиная с максимального по модулю. |
|  | 1 999 -1600 -3400 -2001 | 999 | Минимальный по модулю элемент |

## Выводы.

Была изучена работа и применение условий, циклов и операторов языка C, созданы и применены собственные функции.

Разработана и разбита на функции программа, выполняющая считывание с клавиатуры исходных данных и команды пользователя, выполнение этой команды над массивом и возвращение результата. Была подключена библиотека *stdlib*, из которой функция *abs()* использована для нахождения модуля. Для обработки команд пользователя использовался условный оператор *switch*.

# Приложение А Исходный код программы

Название файла: lb1\_v5.c

#include <stdio.h>

#include <stdlib.h>

#define N 100

int abs\_max(int \*arr, int real\_n){

int a\_max = abs(arr[0]);

int elem = arr[0];

for(int i=0; i<real\_n; i++){

if(abs(arr[i])>a\_max){

a\_max = abs(arr[i]);

elem = arr[i];

}

}

return elem;

}

int abs\_min(int \*arr, int real\_n){

int a\_min = abs(arr[0]);

int elem = arr[0];

for(int i=0; i<real\_n; i++){

if(abs(arr[i])<a\_min){

a\_min = abs(arr[i]);

elem = arr[i];

}

}

return elem;

}

int diff(int \*arr, int real\_n){

int e\_max = abs\_max(arr, real\_n);

int e\_min = abs\_min(arr, real\_n);

return e\_max-e\_min;

}

int sum(int \*arr, int real\_n){

int e\_max = abs\_max(arr, real\_n);

int answer = 0;

for(int i=0;i<real\_n;i++){

if((arr[i]==e\_max)||(e\_max==0)){

answer += arr[i];

e\_max = 0;

}

}

return answer;

}

int main(){

int arr[N];

int real\_n = 0;

int option;

char c;

scanf("%d", &option);

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

scanf("%d%c", &arr[i], &c);

real\_n++;

if(c=='\n'){

break;

}

}

switch (option){

case 0:

printf("%d\n", abs\_max(arr, real\_n));

break;

case 1:

printf("%d\n", abs\_min(arr, real\_n));

break;

case 2:

printf("%d\n", diff(arr, real\_n));

break;

case 3:

printf("%d\n", sum(arr, real\_n));

break;

default:

printf("Данные некорректны\n");

}

return 0;

}