МИНОБРНАУКИ РОССИИ
САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ
ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)
Кафедра МО ЭВМ

ОТЧЕТ

по лабораторной работе №2
по дисциплине «Программирование»
Тема: «Создание make-файла»

Студент гр. 9383 _____ Корсунов А. А.

Преподаватель _____ Жангиров Т. Н.

Санкт-Петербург
2019

Цель работы.

Научиться работать с утилитой **make**.

Задание.
Вариант 6.
В текущей директории создайте проект с make-файлом. Главная цель должна приводить к сборке проекта. Файл, который реализует главную функцию, должен называться menu.c; исполняемый файл - menu. Определение каждой функции должно быть расположено в отдельном файле, название файлов указано в скобках около описания каждой функции.

Реализуйте функцию-меню, 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 отрицательного элемента. (index_first_negative.c)
1 : индекс последнего отрицательного элемента. (index_last_negative.c)
2 : Найти сумму модулей элементов массива, расположенных от первого отрицательного элемента (включая элемент) и до последнего отрицательного (не включая элемент). (sum_between_negative.c)
3 : Найти сумму модулей элементов массива, расположенных до первого отрицательного элемента (не включая элемент) и после последнего отрицательного (включая элемент). (sum_before_and_after_negative.c)

иначе необходимо вывести строку "Данные некорректны".

Выполнение задания.

1) Объявления функций:

объявляются четыре функции, которые требуются в условии задачи.

2) Описание функции main():

*создается массив **a[]** типа int из ста элементов;

*создается переменная **N** со значением 0 типа int (N считает длину массива);

*создается переменная **func** (в ней будут храниться значения от 0 до 3);

*создается переменная **prb** типа char (будет служить для ввода пробелов между значениями массива);

*вводится значение func и пробел;

*с помощью цикла while происходит заполнение массива с условием prb != '\n';

*с помощью оператор switch печатается значение функции в зависимости от значения func;

3) Описание функции index_first_negative(int znc[], int len):

Данная функция принимает на вход значения элементов массива и его длину, с помощью цикла for ищет на интервале от i=0 до len-1 индекс первого отрицательного элемента в массиве и возвращает его, если такого элемента в массиве нет, то возвращает -1;

4) Описание функции last_first_negative(int znc[], int len):

Данная функция принимает на вход значения элементов массива и его длину, с помощью цикла for ищет на интервале от i=len-1 до 0 индекс последнего (первого с конца) отрицательного элемента в массиве и возвращает его, если такого элемента в массиве нет, то возвращает -1;

5) Описание функции sum_between_negative(int znc[], int len):

Данная функция принимает на вход значения элементов массива и его длину, с помощью цикла for ищет на интервале от i = значения первого отрицательного элемента в массиве, которое находится с помощью использования функции, до значения последнего отрицательного элемента-1 и вычисляет сумму элементов, входящих в интервал;

6) Описание функции sum_before_and_after_negative(int znc[], int len):

Данная функция принимает на вход значения элементов массива и его длину, с помощью цикла for считает сумму всех элементов массива до функции **index_first_negative(int znc[], int len)**, а следующим for-ом считает сумму всех элементов массива от **last_first_negative(int znc[], int len)** до конца массива.

Выводы.

В ходе проделанной работы была изучена утилита make, а также были написаны функции, выполняющие определенные задачи.

№ п/п	Входные данные	Выходные данные	Комментарии
1.	0 1 -8 5 -4 5	1	index_first_negative
2.	1 6 6 5 -4 -3	4	index_last_negative
3.	2 1 16 2 -18 -22 15	18	sum_between_negative
4.	3 1 16 2 -18 -22 15 -3	22	sum_before_and_after_negative
5.	6 1 16 2	Данные некорректны	

ПРИЛОЖЕНИЕ А
ИСХОДНЫЙ КОД ПРОГРАММЫ

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

int index_first_negative(int znc[], int len);
int index_last_negative(int znc[], int len);
int sum_between_negative(int znc[], int len);
int sum_before_and_after_negative(int znc[], int len);

int main(){
   int a[100];
   int N=0;
   int func;
   char prb;
   scanf("%d%c", &func, &prb);
   while(prb != '\n') {
      scanf("%d%c", &a[N], &prb);
      N++;
   }
   switch(func)  {
      case 0:
          printf("%d\n", index_first_negative(a, N));
          break;
      case 1:
          printf("%d\n", index_last_negative(a, N));
          break;
      case 2:
         printf("%d\n", sum_between_negative(a, N));
         break;
```

```
        case 3:
            printf("%d\n", sum_before_and_after_negative(a, N));
            break;
        default:
            printf("Данные некорректны\n");
            break;
    }
    return 0;
}

int index_first_negative(int znc[], int len){
    int i = 0;
    for (i; i < len; i++){
        if (znc[i] < 0){
            return i;
        }
     }
     return -1;
}

int index_last_negative(int znc[], int len){
    int i = len - 1;
    for (i; i >= 0; i--){
        if (znc[i] < 0){
            return i;
        }
    }
     return -1;
}
```

```
int sum_between_negative(int znc[], int len) {
    int sum_negative = 0;
    int last_negative = index_last_negative(znc, len);
    int i = index_first_negative(znc, len);
    for (i; i < last_negative; i++){
        sum_negative += abs(znc[i]);
    }
    return sum_negative;
}

int sum_before_and_after_negative(int znc[], int len) {
    int first_negative = index_first_negative(znc, len);
        int last_negative = index_last_negative(znc, len);
        int i = 0;
        for(i; i < first_negative; i++){
        sum_between += abs(znc[i]);
        }
        for(last_negative; last_negative < len; last_negative++){
                sum_between += abs(znc[last_negative]);
        }
        return sum_between;
}
```