**ОТЧЕТ**

по лабораторной работе №6

по дисциплине «Программирование»

на тему «Макросы»

Вариант №4

Выполнили:

студенты группы 23ВВВ4

Зарубин Я.Д.

Костин К.

Приняли:

Голотенков Н.О.

к.т.н. доцент Слепцов Н.В.

**Название**

Макросы

**Цель работы**

изучение аппарата макросредств ассемблера и получение навыков разработки и использования макроопределений.

**Лабораторное задание**

Разработать программу вычисления таблицы значений функции *y=f(a,b,c,d,e)* при изменении одного из ее аргументов в соответствии с заданной зависимостью. Обеспечить вывод на экран вида табулируемой функции, значений неизменяемых аргументов и результатов вычислений в виде пар: “значение варьируемого аргумента – значение функции”.

**Описание метода решения задачи**

Эта программа решает задачу вычисления выражения y = a + b \* c / d - e для каждого элемента массива d и сохранения результатов в массив y. Для этого используется вставка ассемблерного кода в среде программирования на Си. В программе определены три макроса: f1, f2 и func.

1. Макрос f1 вычисляет значения элементов массива d по формуле 4 \* i + 1, где i - индекс элемента массива.

2. Макрос f2 вычисляет значения элементов массива d по формуле 5 \* i + 4, где i - индекс элемента массива.

3. Макрос func вычисляет значения элементов массива y по формуле y = a + b \* c / d - e, используя значения из массивов a, b, c, d и e.

В функции main() определяются начальные значения переменных и массивов, после чего выполняется вставка ассемблерного кода для вычисления значений массивов d и y. Результаты выводятся на экран с помощью функции printf().

### Листинг

// y=a+b\*c/d-e  
//4 di i+4\*a 5\*i+3  
  
[#include](https://vk.com/im?sel=515315317&st=%23include) <stdio.h>  
[#include](https://vk.com/im?sel=515315317&st=%23include) <conio.h>  
  
[#define](https://vk.com/im?sel=515315317&st=%23define) f1(a, edi) {

\_\_asm mov ecx, 0\  
\_\_asm start1:\  
\_\_asm mov eax, a\  
\_\_asm imul eax, 4\  
\_\_asm add eax, ecx\  
\_\_asm add eax, 1\  
\_\_asm mov[edi + ecx \* 4], eax\  
\_\_asm inc ecx\  
\_\_asm cmp ecx, 5\  
\_\_asm jne start1}  
  
[#define](https://vk.com/im?sel=515315317&st=%23define) f2(edi) {

\_\_asm start2:\  
\_\_asm mov eax, ecx\  
\_\_asm imul eax, 5\  
\_\_asm add eax, 4\  
\_\_asm mov[edi + ecx \* 4], eax\  
\_\_asm inc ecx\  
\_\_asm cmp ecx, 10\  
\_\_asm jne start2}  
  
[#define](https://vk.com/im?sel=515315317&st=%23define) func(a, b, c, edi, e, esi) {

\_\_asm xor ecx, ecx\  
\_\_asm startf :\  
\_\_asm mov eax, b\  
\_\_asm imul eax, c\  
\_\_asm mov ebx, [edi + ecx \* 4]\  
\_\_asm cdq\  
\_\_asm idiv ebx\  
\_\_asm add eax, a\  
\_\_asm sub eax, e\  
\_\_asm mov[esi + ecx \* 4], eax\  
\_\_asm inc ecx\  
\_\_asm cmp ecx, 10\  
\_\_asm jne startf}  
  
int main() {  
int a = 10

b = 1

c = 1

d[10]

e = 3

y[10]; // результаты вычисления формулы

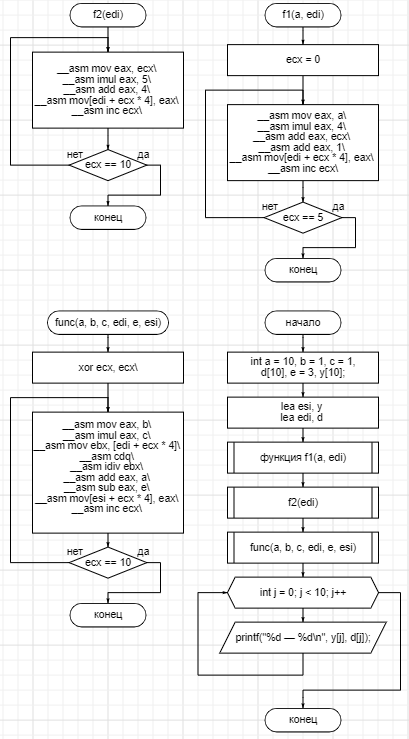
if (a<=0) { //условие считается положительным, если а больше и не равно 0.

printf("Ошибка, переменная должна быть положительной";

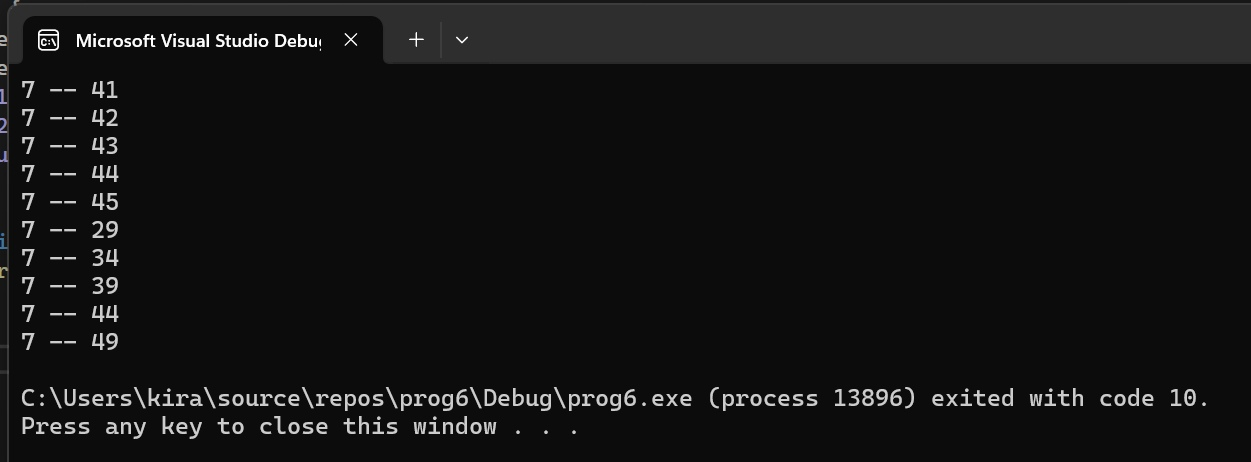
return 1;

}  
\_\_asm {  
lea esi, y  
lea edi, d  
f1(a, edi)  
f2(edi)  
func(a, b, c, edi, e, esi)  
}  
  
for (int j = 0; j < 10; j++) {  
printf("%d — %d\n", y[j], d[j]);  
}  
}

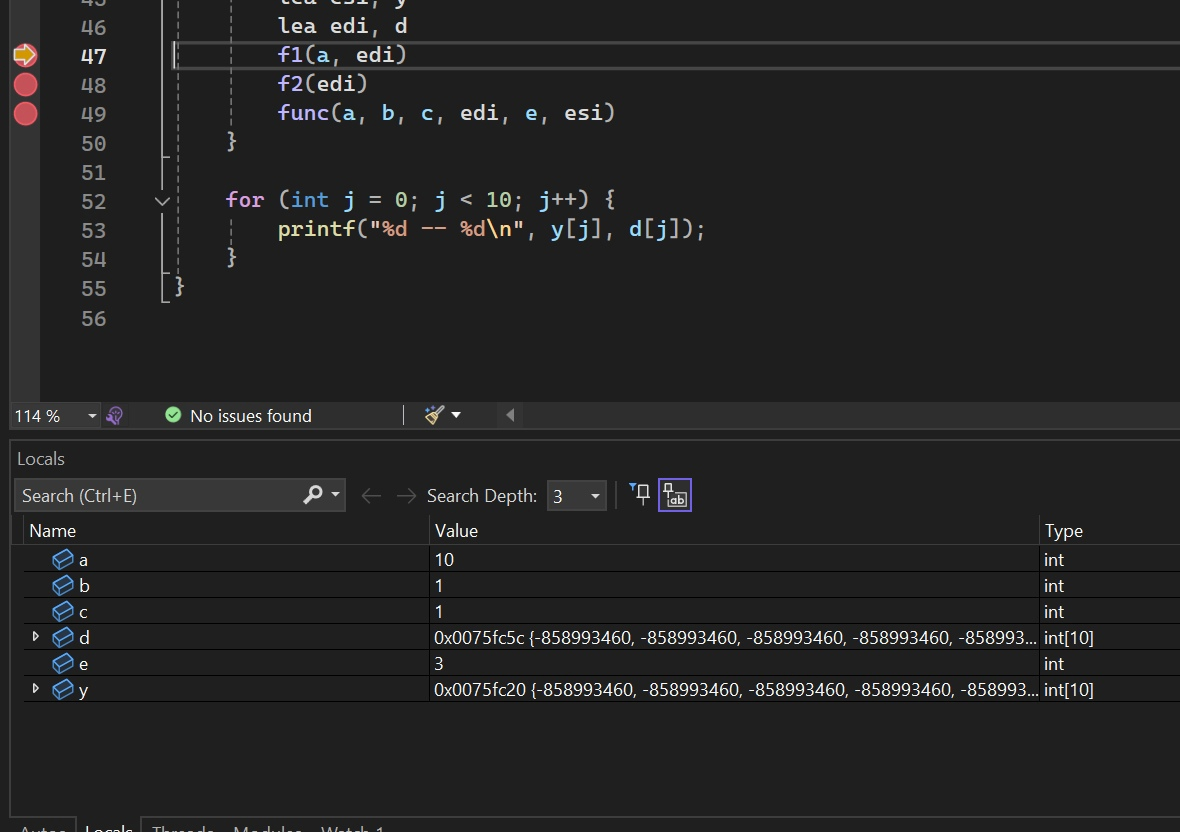
**Схема программы**



**Результат работы программы**



**Рисунок 1 — Результаты работы программы**



**Рисунок 2 — Протокол трассировки**

На рисунке 2 показан протокол трассировки после заполнения данных и вычислений.

Результат работы программы, показанный на рисунке 1, совпал с результатами трассировки.

### Вывод

были изучены аппараты макросредств ассемблера и получены навыки разработки и использования макроопределений.