Цель работы

Освоение процесса отладки программы, вычисляющей заданную функцию.

Задание

Требуется разработать на языке Ассемблер IBM PC программу, которая по заданным целым значениям a, b, i, k вычисляет сначала значения i1 = fn1(a, b, i) и i2 = fn2(a, b, i), а затем значение res = fn3(i1, i2, k). Вид функций fn1, fn2 определяется из табл. 1, а вид функции fn3 - из табл. 2. Символы n1, n2, n3, используемые в описании функции f, соответствуют числовым индексам функции, которые представлены тремя соответствующими цифрами в табл. 3 индивидуальных заданий (n1, n2, n3).

**Вариант 4**

i1 = f2

i2 = f4

res = f2

|  |  |
| --- | --- |
|  | при a > b  при a ≤ b |
|  | при a > b  при a ≤ b |
|  | max (i1, 10-i2), при k<0  ‌|i1‌‌-i2|, при k≥0 |

**Текст программы:**

TITLE LAB5

.Model Small

.Stack 100h

.Data

a db 4

b db 3

i db 2

i1 db ?

i2 db ?

k db 1

res db ?

.Code

begin:

mov ax, @data

mov ds, ax

call ClearAllRegisters

lea di, a

mov al, [di]

lea di, b

cmp al, [di] ; a<=b?

jna a\_is\_less\_or\_equal\_b

; a > b:

; i1 = 5+4\*i

call ClearAllRegisters

mov al, 4

lea di, i

mov cl, [di]

call MultipleNumbers ; dl = 4\*i

add dl, 5 ; dl = 4\*i+5

lea di, i1

mov [di], dl

; i2 = 3\*i-4

call ClearAllRegisters

mov al, 3

lea di, i

mov cl, [di]

call MultipleNumbers ; dl = 3\*i

sub dl, 4 ; dl = 3\*i - 4

lea di, i2

mov [di], dl

jmp get\_results

a\_is\_less\_or\_equal\_b:

; i1= 4\*i-10

call ClearAllRegisters

mov al, 4

lea di, i

mov cl, [di]

call MultipleNumbers ; dl = 3\*i

sub dl, 10 ; dl = 3\*i - 4

lea di, i1

mov [di], dl

; i2=3\*(i+1)

call ClearAllRegisters

mov al, 3

lea di, i

mov cl, [di]

inc cl

call MultipleNumbers ; dl = 3 \* (i+1)

lea di, i2

mov [di], dl

get\_results:

call ClearAllRegisters

lea di, i1

mov al, [di]

lea di, i2

lea si, k

mov bl, [si]

cmp bl, 0

jl k\_less\_0

; k >= 0

; res=|i1-i2|

sub al, [di]; вычитаем i1-i2 и сохраняем в al

test al, al ; проверяем, отрицательность

jns save\_result

neg al ; модуль

jmp save\_result

k\_less\_0:

; k < 0

; res=max (i1, 10-i2)

sub [di], 10;

neg [di]; 10-i2

cmp [di], al ; сравниваем 10-i2 и i1

jl save\_result ;10-i2<i1

mov al, [di] ; 10-i2>i1

save\_result:

; al = res

lea di, res

mov [di], al

exit:

mov ah, 01h

int 21

mov ax,4c00h

int 21h

LOCALS

; умножение ax на cx

MultipleNumbers PROC

test cl, cl ; проверяем, отрицательность множителя

jns @@Sum

neg cl ; модуль

mov bl, 1 ; tmp для сохранения знака

@@Sum:

add dl, al ; dx - результат, ax - множимое

loop @@Sum

cmp bl, 0

je @@end\_procedure

neg dl ; если был сохранён знак, то

@@end\_procedure:

mov ax, 0

ret

MultipleNumbers ENDP

; очистка регистров

ClearAllRegisters PROC

mov ax, 0

mov cx, 0

mov bx, 0

mov dx, 0

mov si, 0

mov di, 0

ret

ClearAllRegisters ENDP

END begin

**Тест 1**

.Data

a db 4

b db 3

i db 2

i1 db ?

i2 db ?

k db 1

res db ?

a = 4, b = 3 => a>b=true

т.о. i1=5+4\*i=13

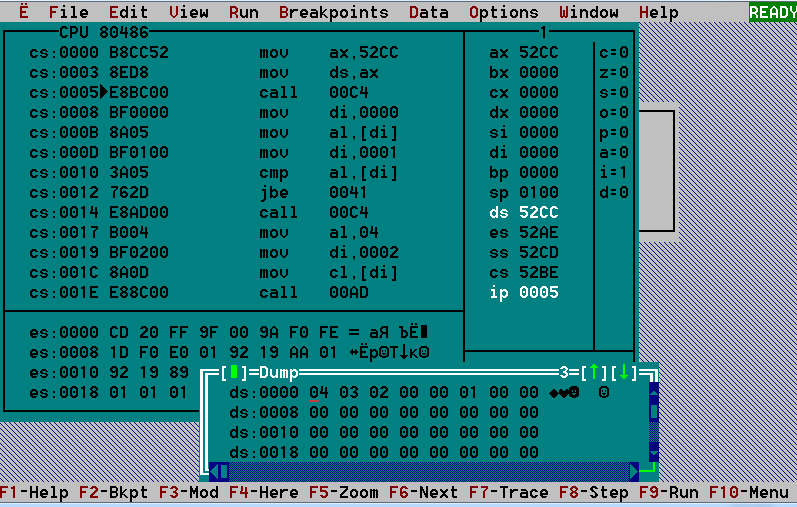


Рис.1.1.1 Начало отладки программы и содержимое data segment

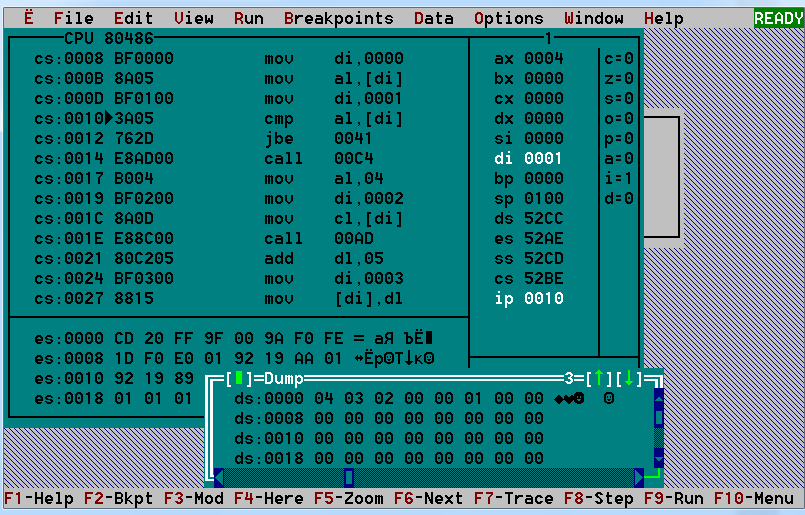


Рис.1.1.2 Сравнение a и b

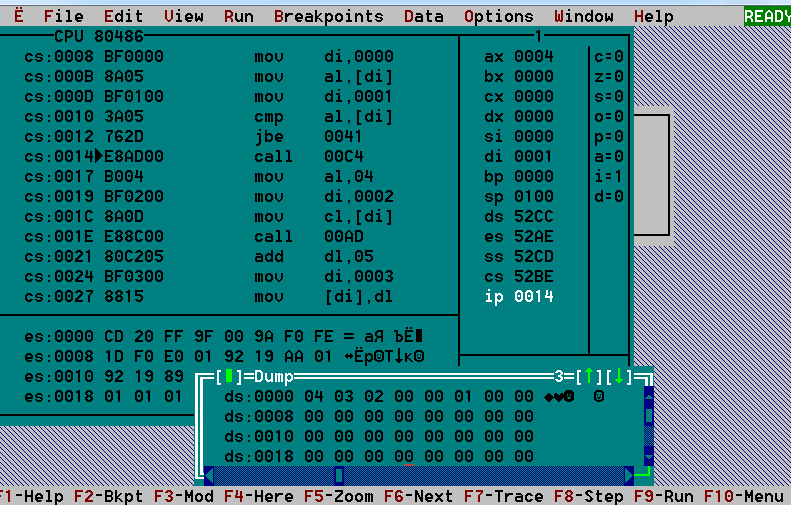


Рис.1.1.3 Продолжение последовательного выполнения программы, поскольку a>b

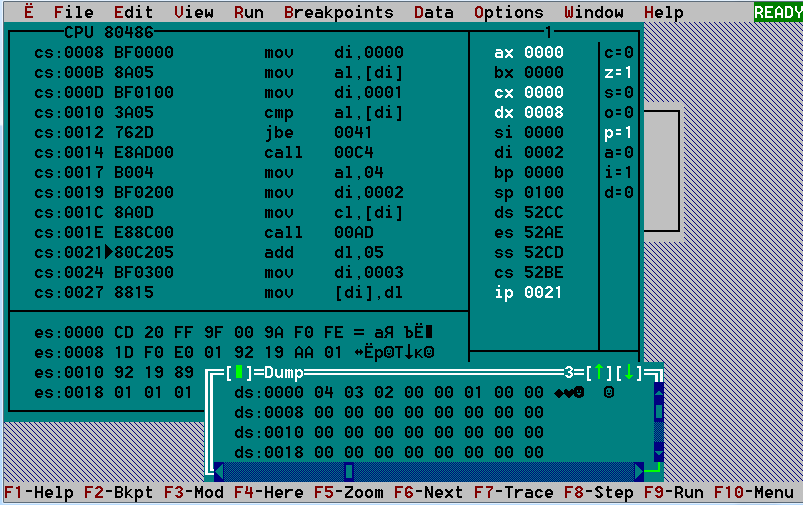


Рис. 1.1.4 Результат умножения i\*4

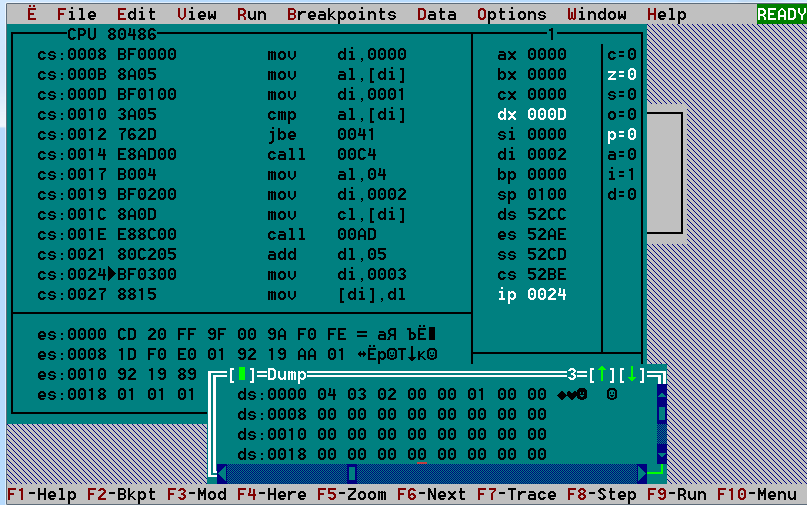


Рис. 1.1.5 Результат сложения 4\*i+5=13 (D16)

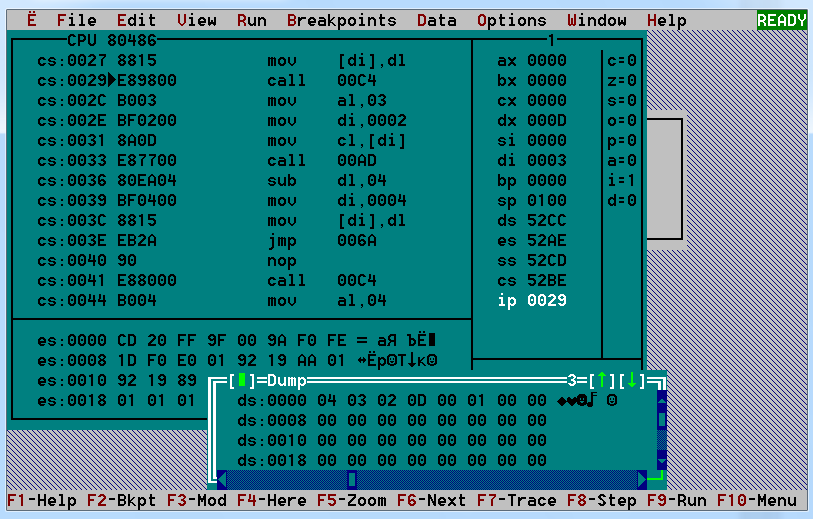


Рис. 1.1.6 Запись результата i1=1310 (D16) в сегмент памяти для i1

a = 4, b = 3 => a>b=true

т.о. i2=3\*i-4=2

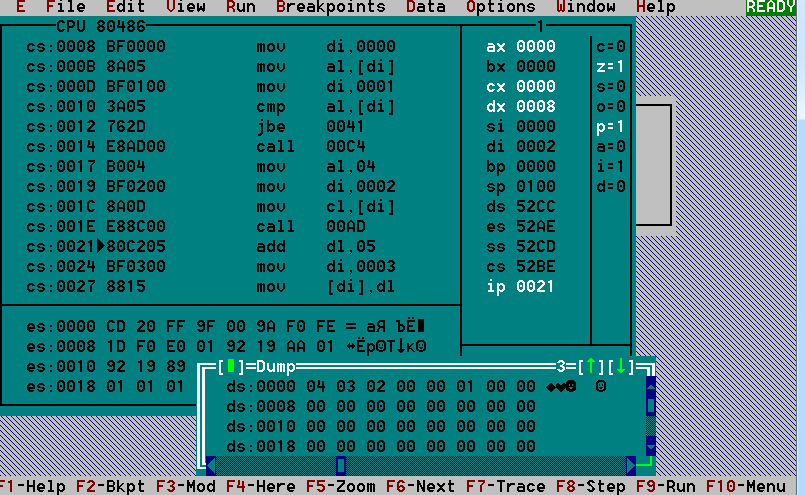


рис.1.2.1 Умножили 3\*i=6

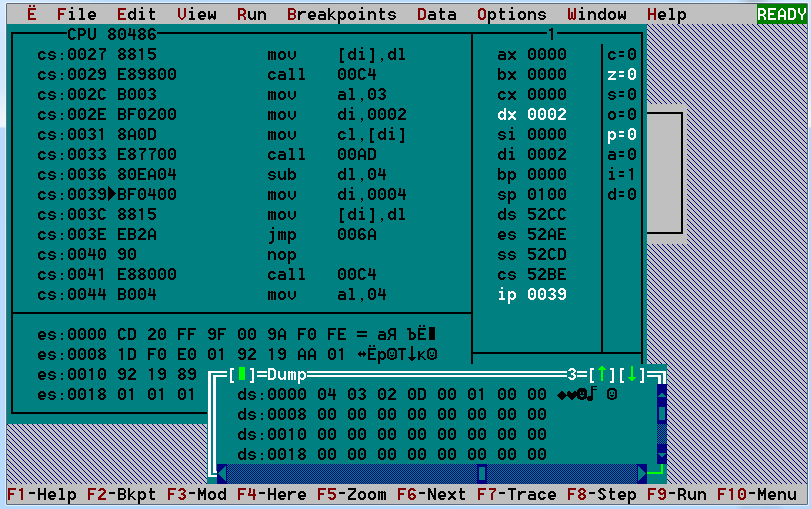


рис.1.2.2 Вычли 4 из 3\*i

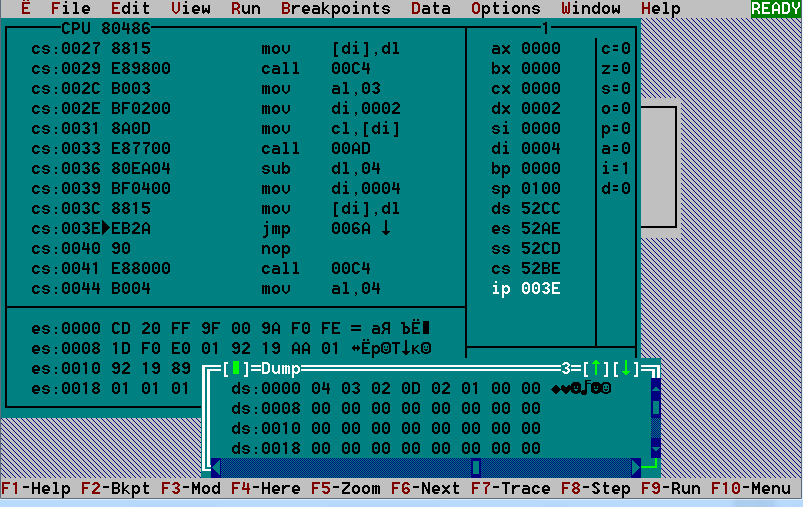


рис.1.2.3 Сохранили i2=2 в сегмент памяти i2

т.к. k=1, то k>0, значит res=|i1-i2|=1110 (B16)

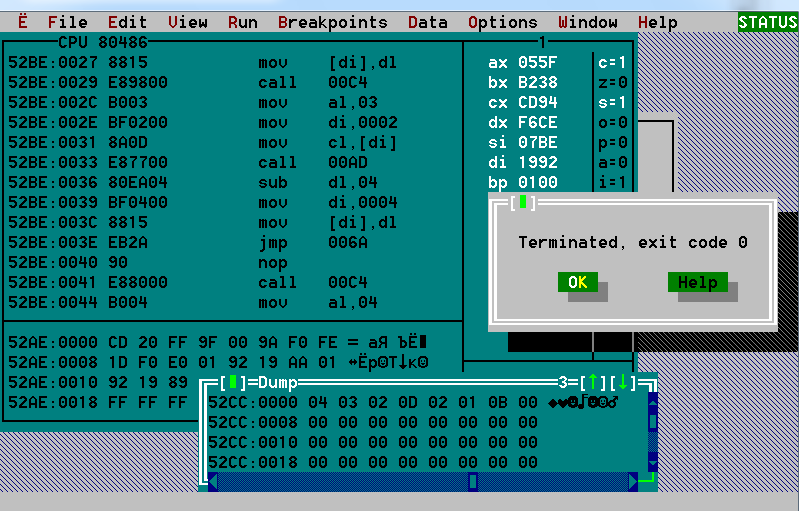


рис.1.3.1 Результат работы программы записан в сегмент памяти res

**Тест 2**

.Data

a db 3

b db 4

i db 2

i1 db ?

i2 db ?

k db 1

res db ?

т.о.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| a | b | i | k | i1 | i2 | res |
| 3 | 4 | 2 | 1 | -2 | 9 | 11 |

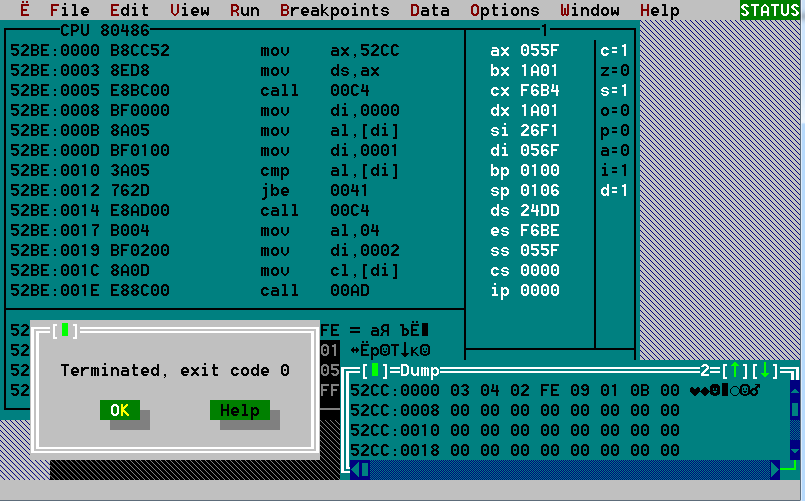


Рис.2 Результат работы программы (1110 = 0B16)

**Тест 3**

.Data

a db 4

b db 3

i db 2

i1 db ?

i2 db ?

k db -1

res db ?

т.о.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| a | b | i | k | i1 | i2 | res |
| 4 | 3 | 2 | -1 | 13 | 2 | 13 |

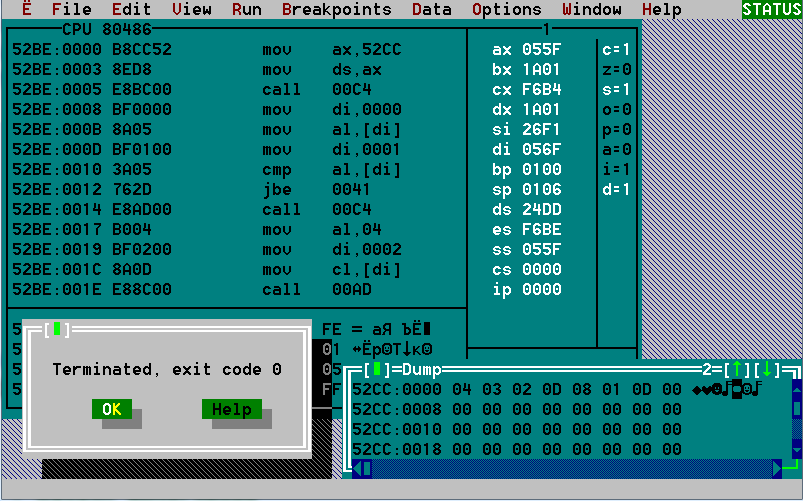


Рис.3 Результат работы программы (1310 = 0D16)

**Тест 4**

.Data

a db 3

b db 4

i db 2

i1 db ?

i2 db ?

k db -1

res db ?

т.о.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| a | b | i | k | i1 | i2 | res |
| 3 | 4 | 2 | -1 | -2 | 9 | 1 |

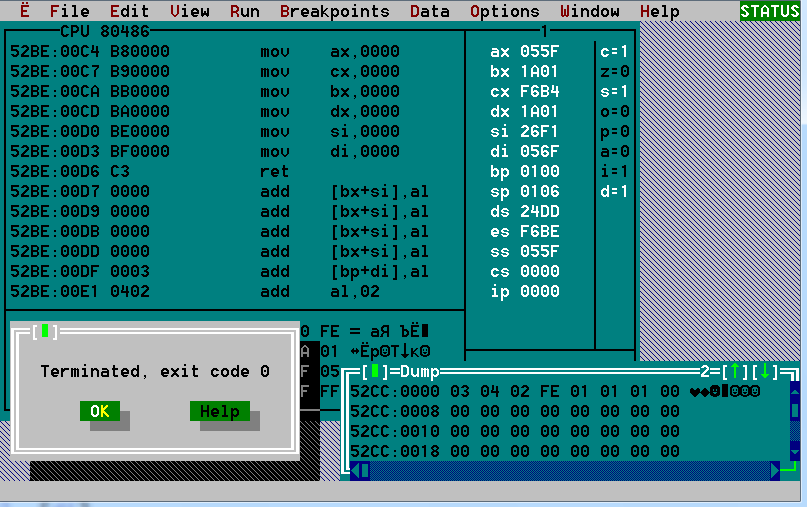


Рис.4 Результат работы программы (110 = 0116)

**Тест 5**

.Data

a db 4

b db 4

i db 2

i1 db ?

i2 db ?

k db 1

res db ?

т.о.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| a | b | i | k | i1 | i2 | res |
| 4 | 4 | 2 | 1 | -2 | 9 | 11 |

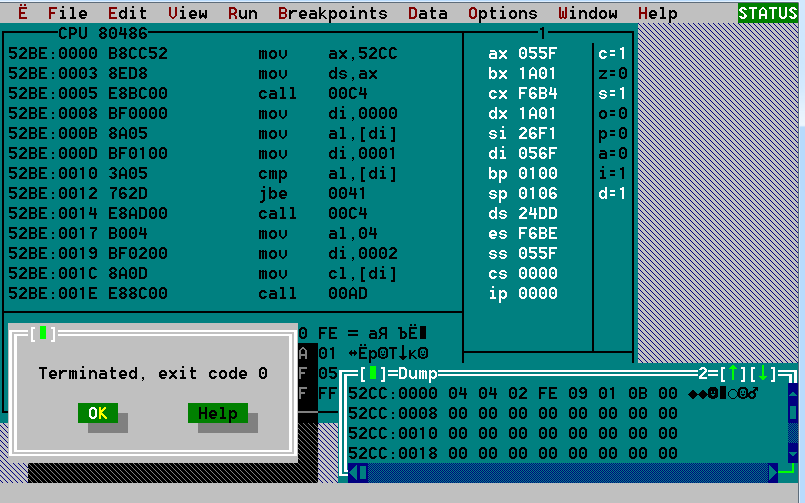


Рис.5 Результат работы программы (1110 = 0B16)