# Comp 3350: Computer Organization & Assembly Language

# HW # 5: Theme: Data Definitions, Addressing Modes, Arrays

*All main questions carry equal weight.*

*(Credit awarded to only those answers for which work has been shown.)*

**Table of Contents:**

* Question 1 is answered on the first page.
* Question 2 is answered on pages 2 – 7.
* Question 3 is answered on pages 8 – 9.
* Question 4 is answered on page 10.

1. [Memory Map] Fill in the following memory diagram with the data provided below. Please assume that the data segment begins at 0x0040700A.

.data

Alpha WORD 34CDh, 0ABh

Beta BYTE 78h

Gamma DWORD 0AACD1134h

Delta BYTE 5Ch

|  |  |  |
| --- | --- | --- |
| *Address* | *Variable* | *Data* |
| 0040700A | ***Alpha*** | ***CDh*** |
| 0040700B | ***Alpha*** | ***34h*** |
| 0040700C | ***Alpha*** | ***ABh*** |
| 0040700D | ***Alpha*** | ***00h*** |
| 0040700E | ***Beta*** | ***78h*** |
| 0040700F | ***Gamma*** | ***34h*** |
| 00407010 | ***Gamma*** | ***11h*** |
| 00407011 | ***Gamma*** | ***CDh*** |
| 00407012 | ***Gamma*** | ***AAh*** |
| 00407013 | ***Delta*** | ***5Ch*** |

1. [Addressing Modes] Copy the following code into your assembly development environment and single-step through it. For each single step execution, submit the screenshot. For those instructions referencing memory, do the linear address computation (typewritten/handwritten).

TITLE Addressing Modes (main.asm)

INCLUDE Irvine32.inc

.data

alpha DWORD 70405046h, 50607080h

beta DWORD 91A3B2C7h, 0D0E0F012h

gamma DWORD 0CDABh

.code

main PROC

mov eax, A1C3h; Immediate

mov ecx, eax; Register to Register

mov edi, OFFSET beta; Immediate

mov [gamma], eax; Direct

mov esi, gamma; Direct

mov esi, 4; Immediate

mov eax, beta[esi]; Indirect-offset

mov ebx, OFFSET alpha; Immediate

mov eax, [ebx]; Indirect

mov eax, 4[ebx]; Indirect-displacement

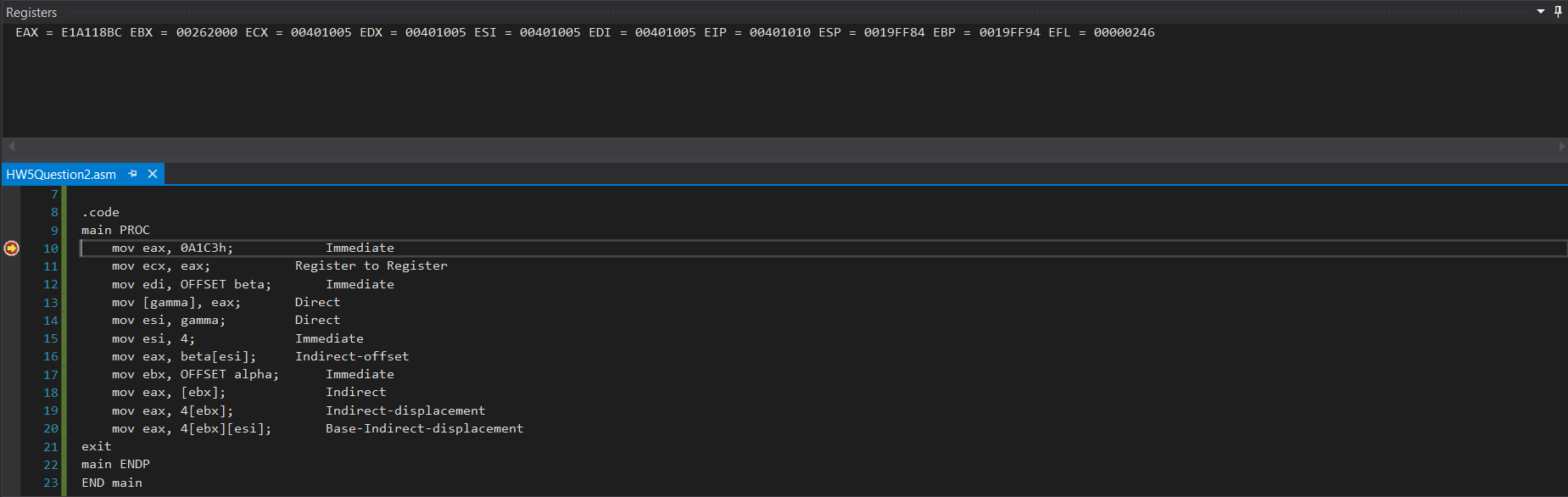
mov eax, 4[ebx][esi]; Base-Indirect-displacement

exit

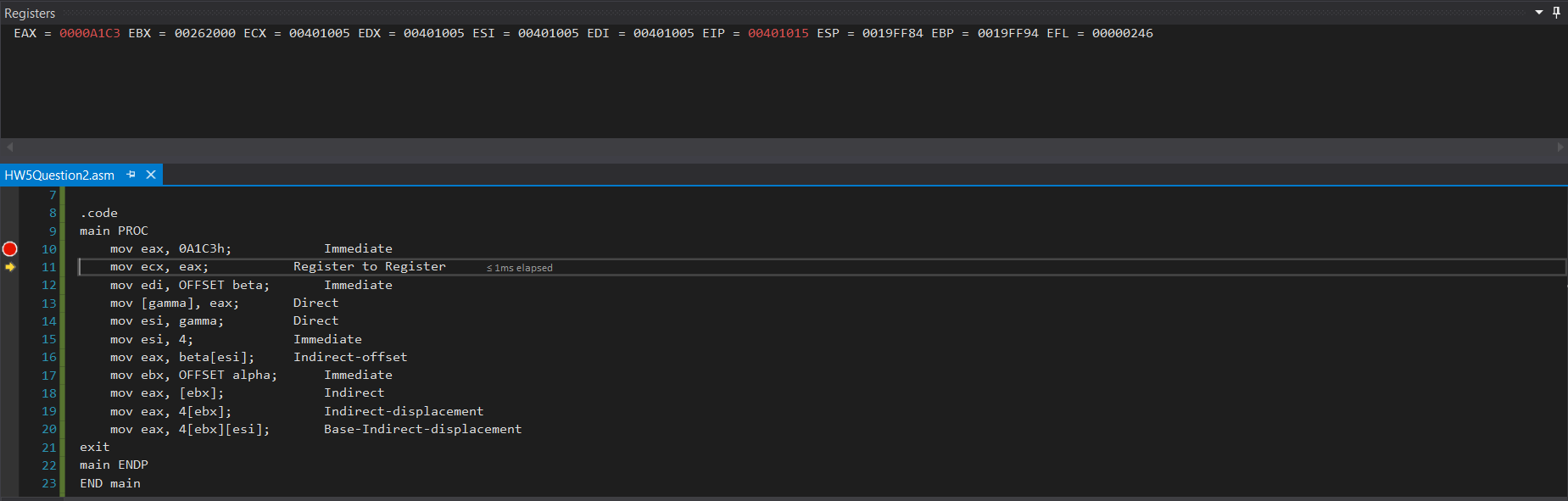
main ENDP

END main

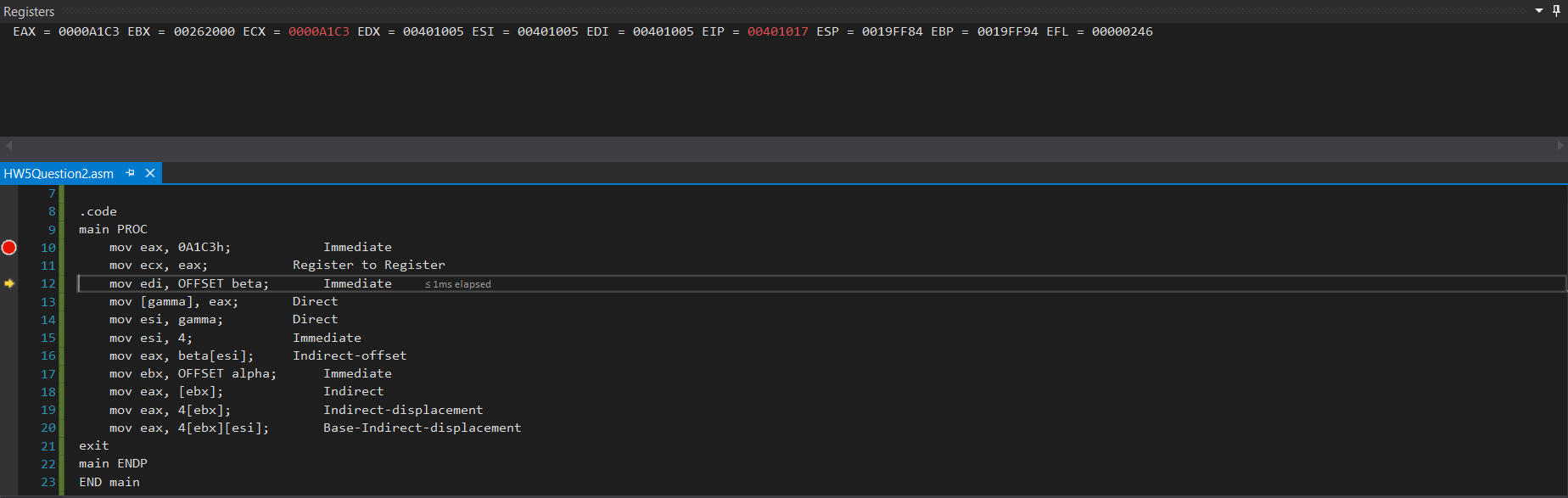
Step 1:



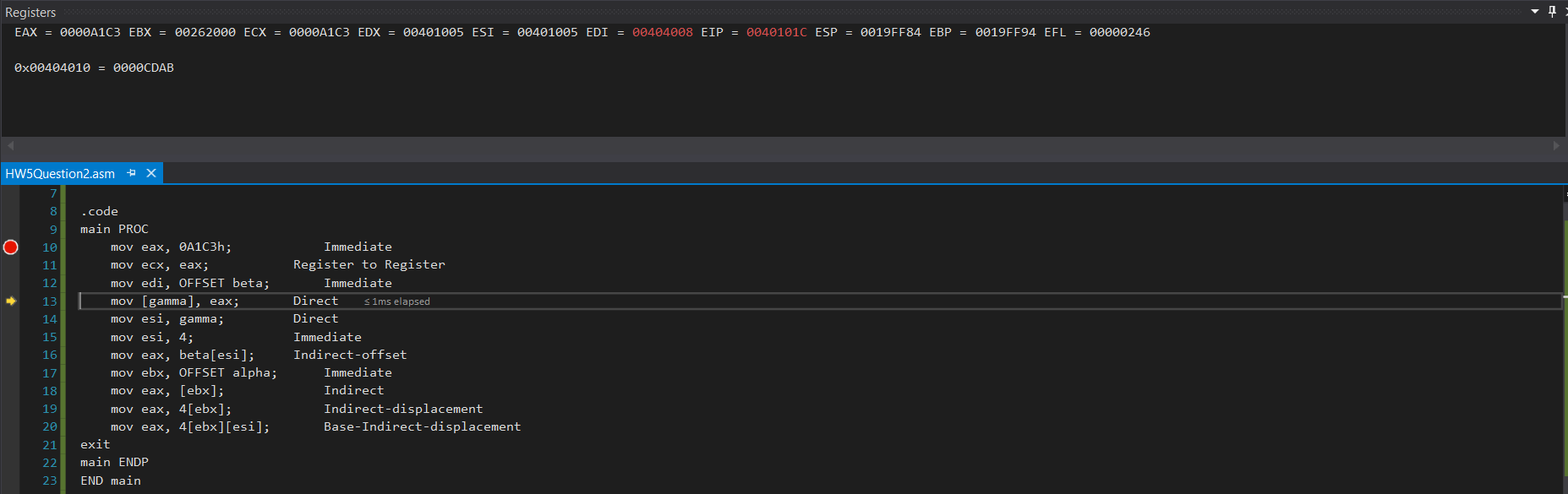
Step 2:



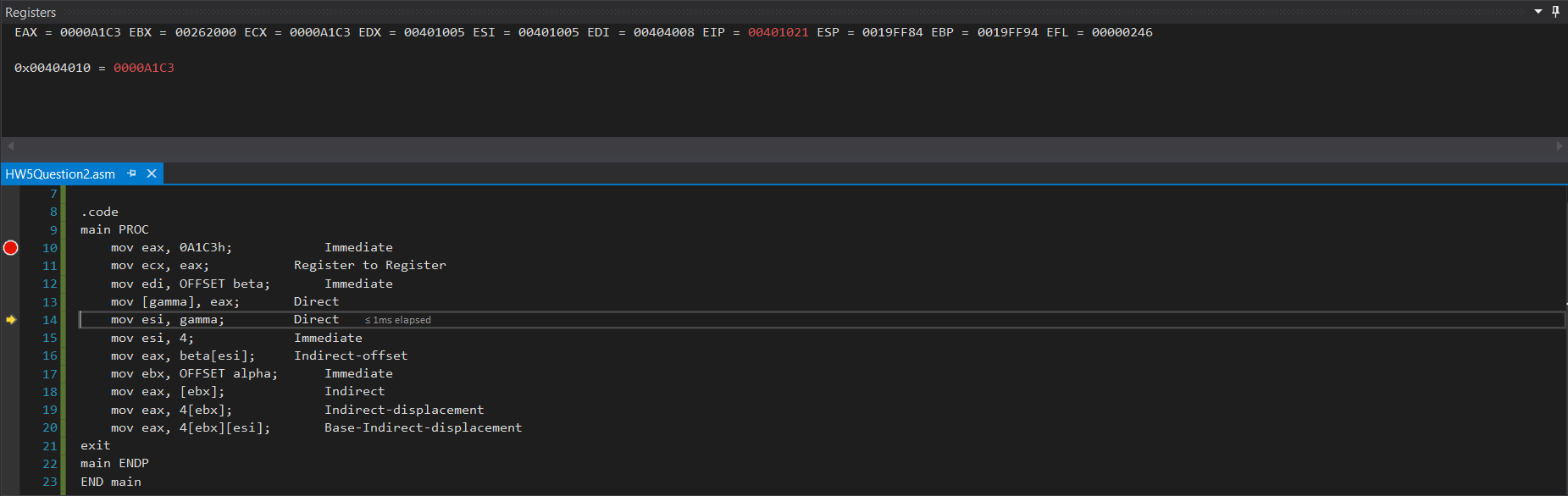
Step 3:



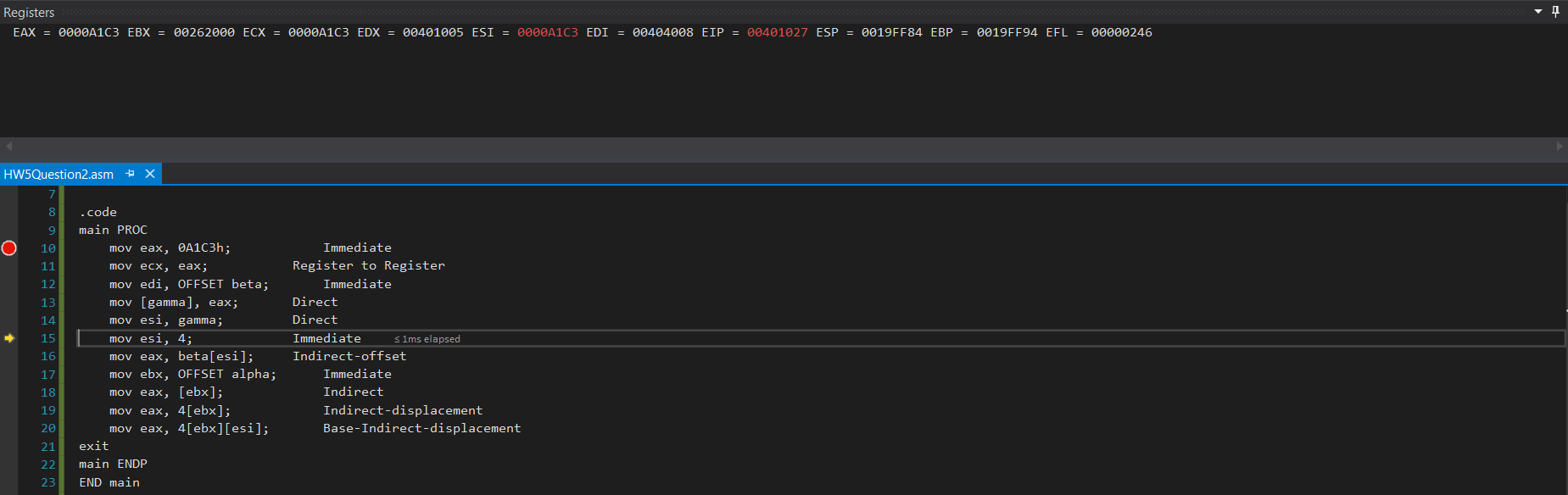
Step 4:



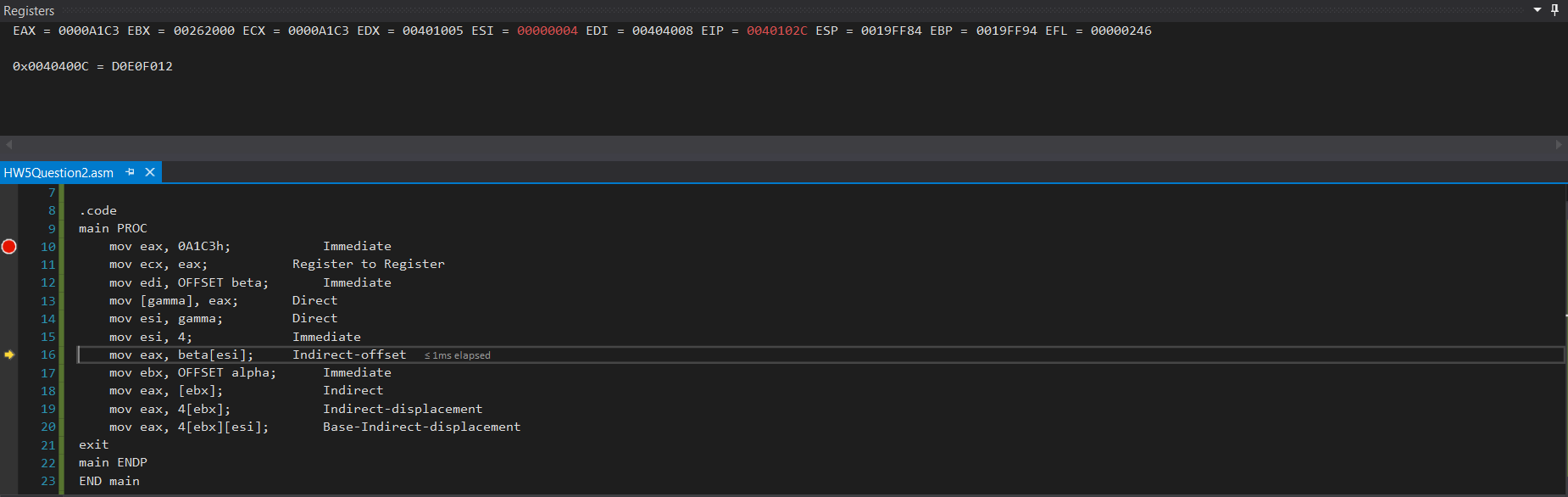
Step 5:



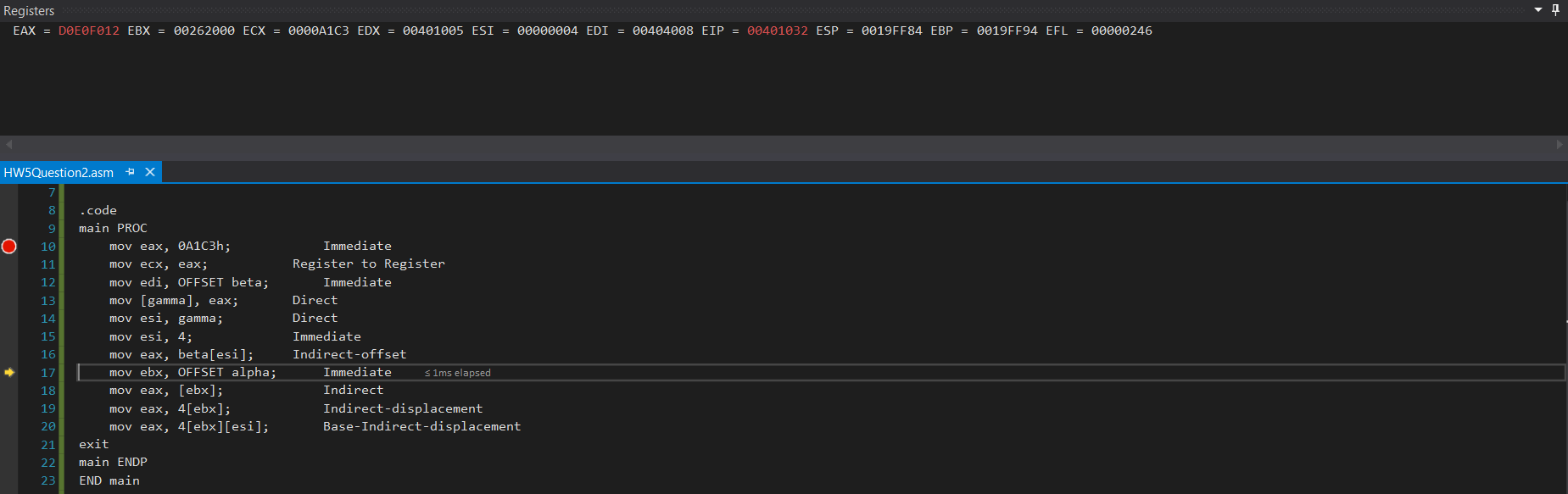
Step 6:



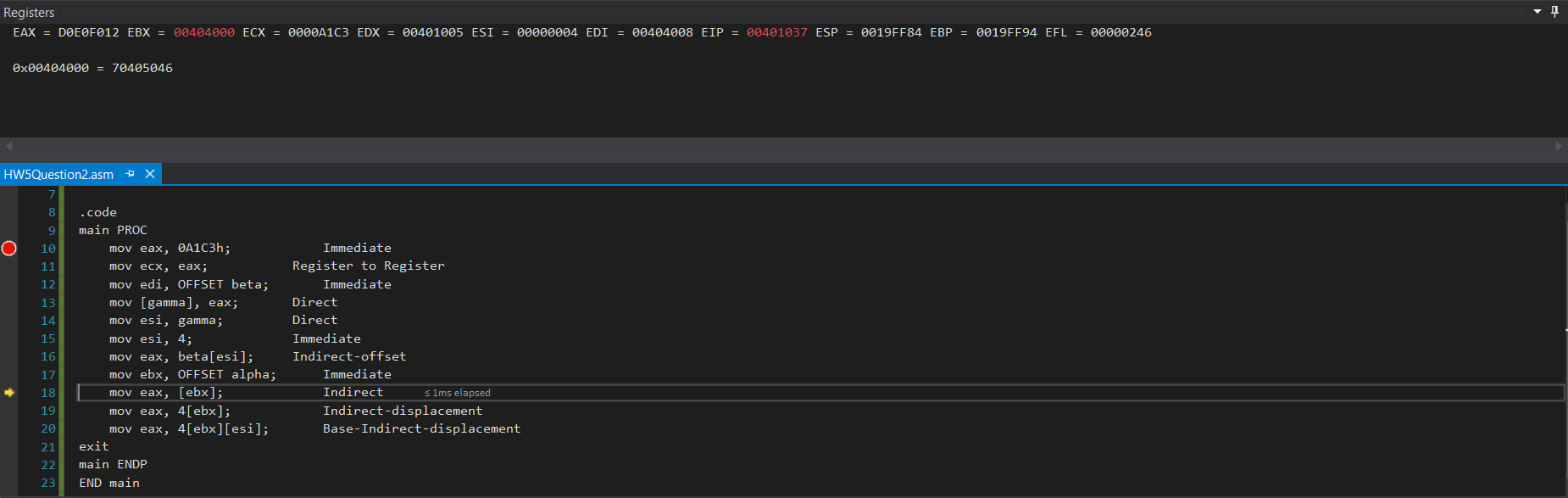
Step 7:



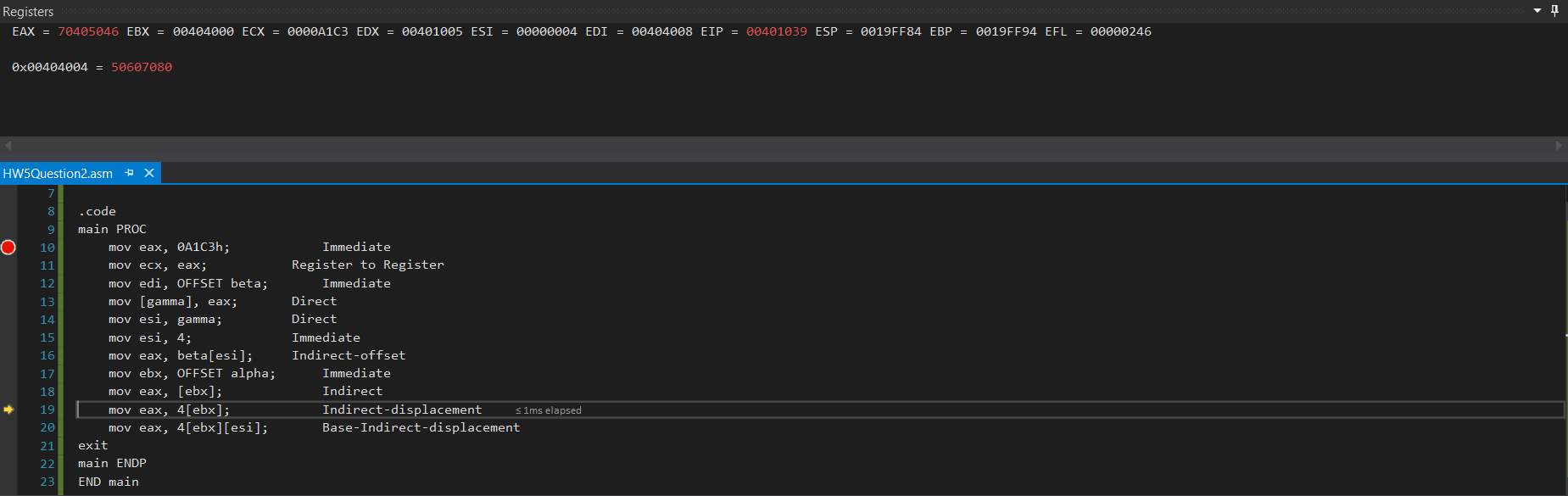
Step 8:



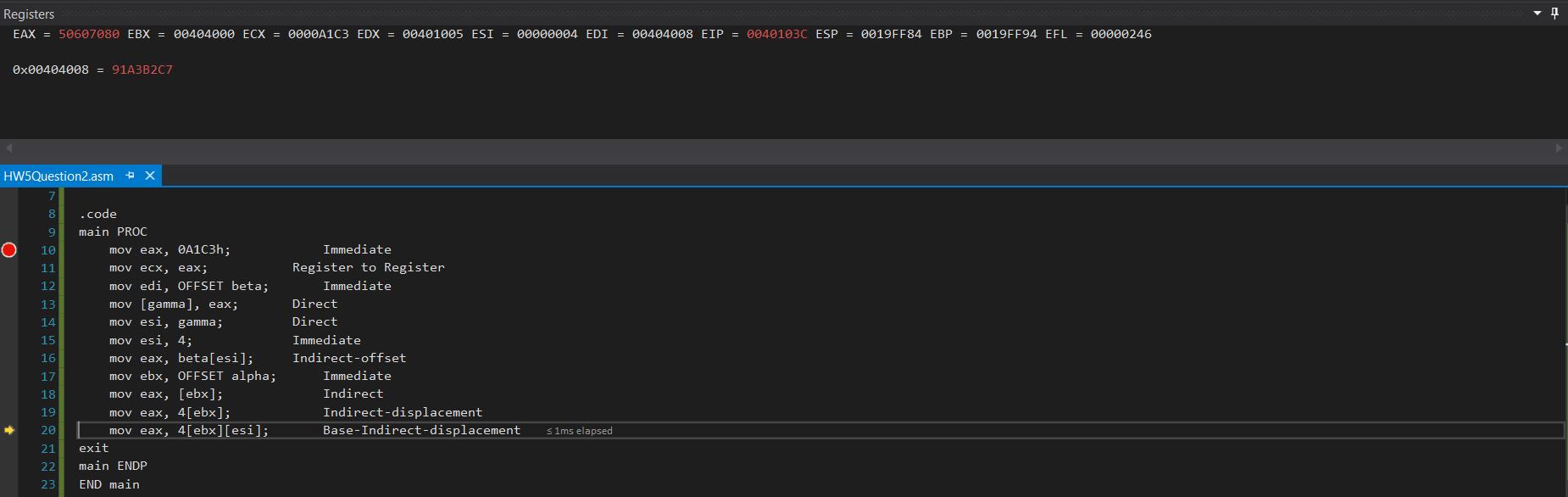
Step 9:



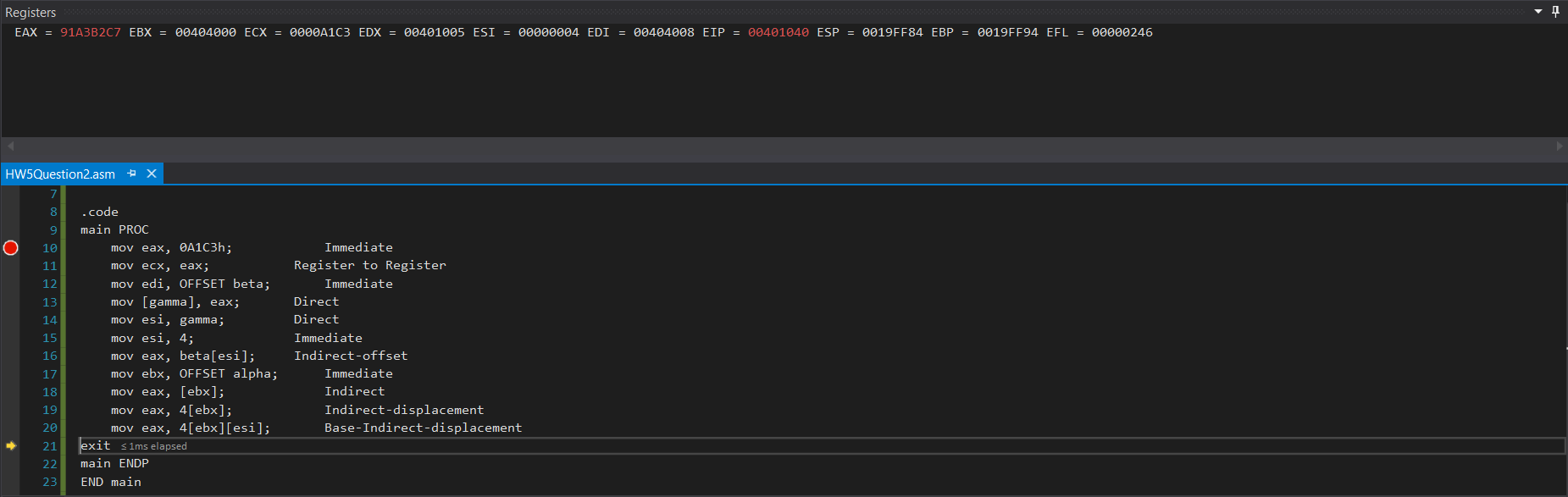
Step 10:



Step 11:



Step 12:



**Linear Addresses for Each:**

mov eax, A1C3h; Immediate

mov ecx, eax; Register to Register

mov edi, OFFSET beta; Immediate

mov [gamma], eax; Direct gamma; 0x00404016 = 0A1C3h

mov esi, gamma; Direct gamma; esi = 0A1C3h (0x00404016)

mov esi, 4; Immediate

mov eax, beta[esi]; Indirect-offset beta + 4; 00404012 (0x00404012)

mov ebx, OFFSET alpha; Immediate

mov eax, [ebx]; Indirect ebx; eax = 70405046h (0x00404000)

mov eax, 4[ebx]; Indirect-displacement 4 + ebx; eax = 50607080h (0x00404004)

mov eax, 4[ebx][esi]; Base-Indirect-displacement 4 + ebx + esi; eax = 91A3B2C7h (0x00404008)

1. [Indirect addressing] Write a program that adds the corresponding even indexed elements of Array1 and Array2 and stores the results in Array3; e.g. for the 8th element, Array3 [7] 🡨 Array1 [7] + Array2 [7]. Note that Array3 will have about half the number of elements of the other two arrays. Include commands to display the elements of all the arrays. Submit screenshot of the displays of the elements of all the arrays. You can use WriteInt or WriteHex to display the elements of the arrays. Fill in Array1 and Array2 each by your own ten numbers.

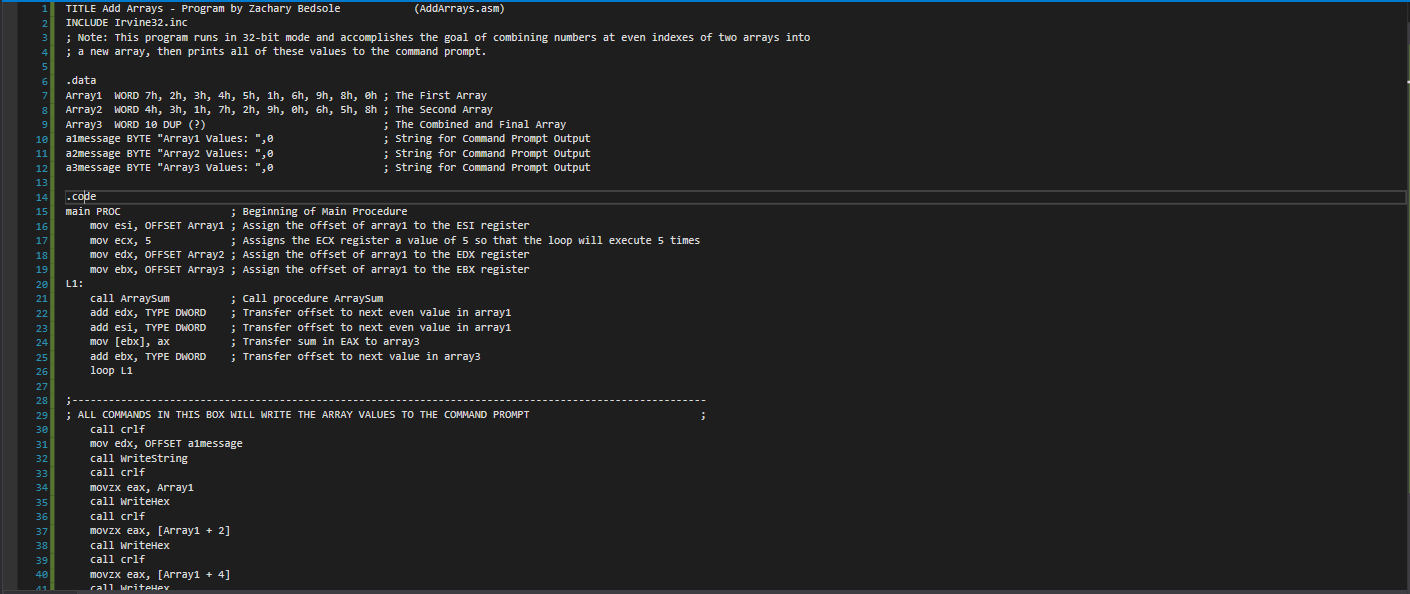
.data

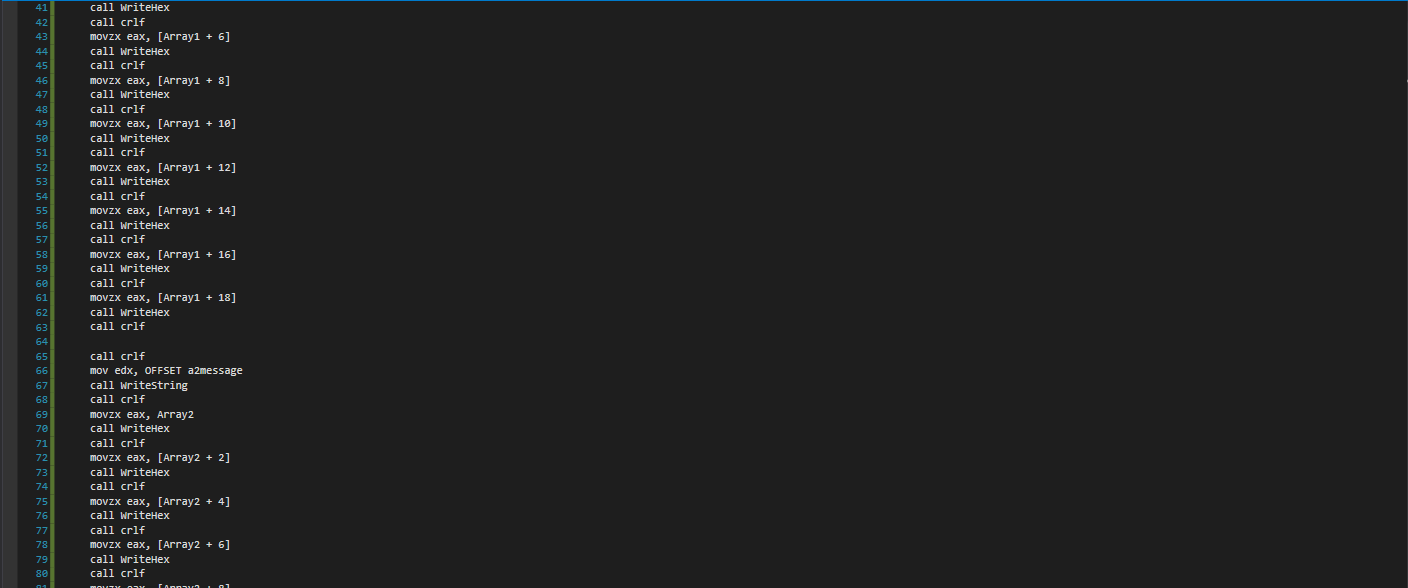
Array1 WORD 7h, 2h, 3h, 4h, 5h, 1h, 6h, 9h, 8h, 0h

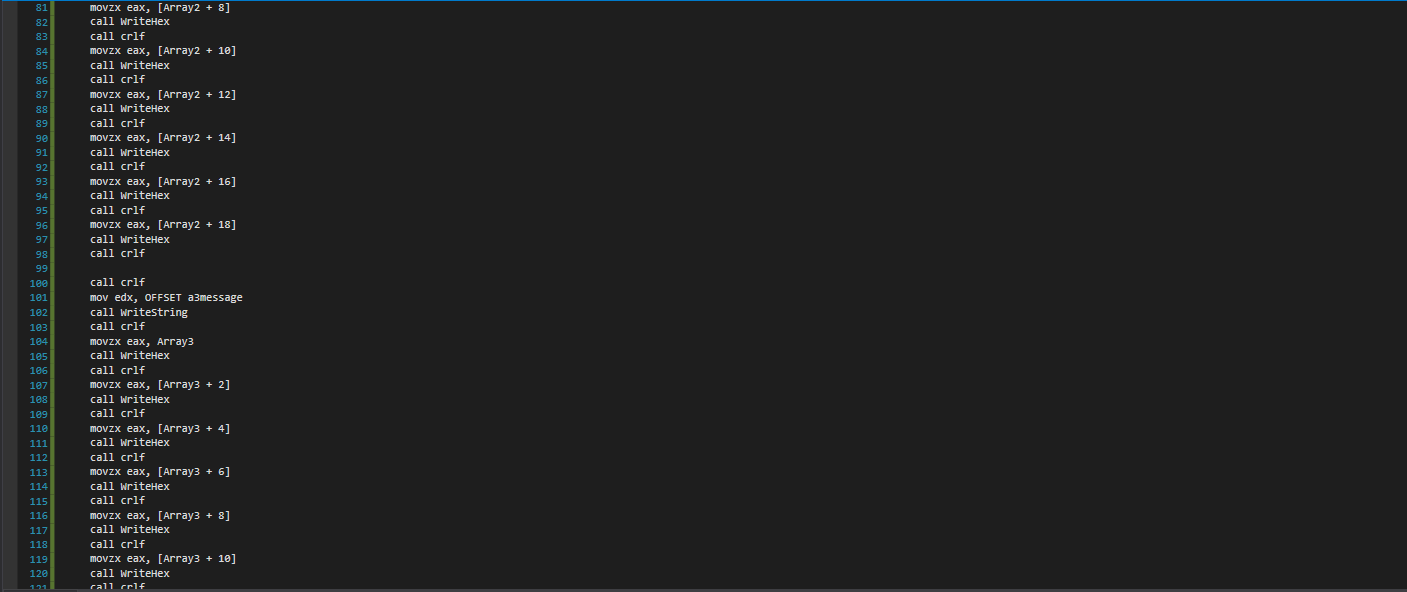
Array2 WORD 4h, 3h, 1h, 7h, 2h, 9h, 0h, 6h, 5h, 8h

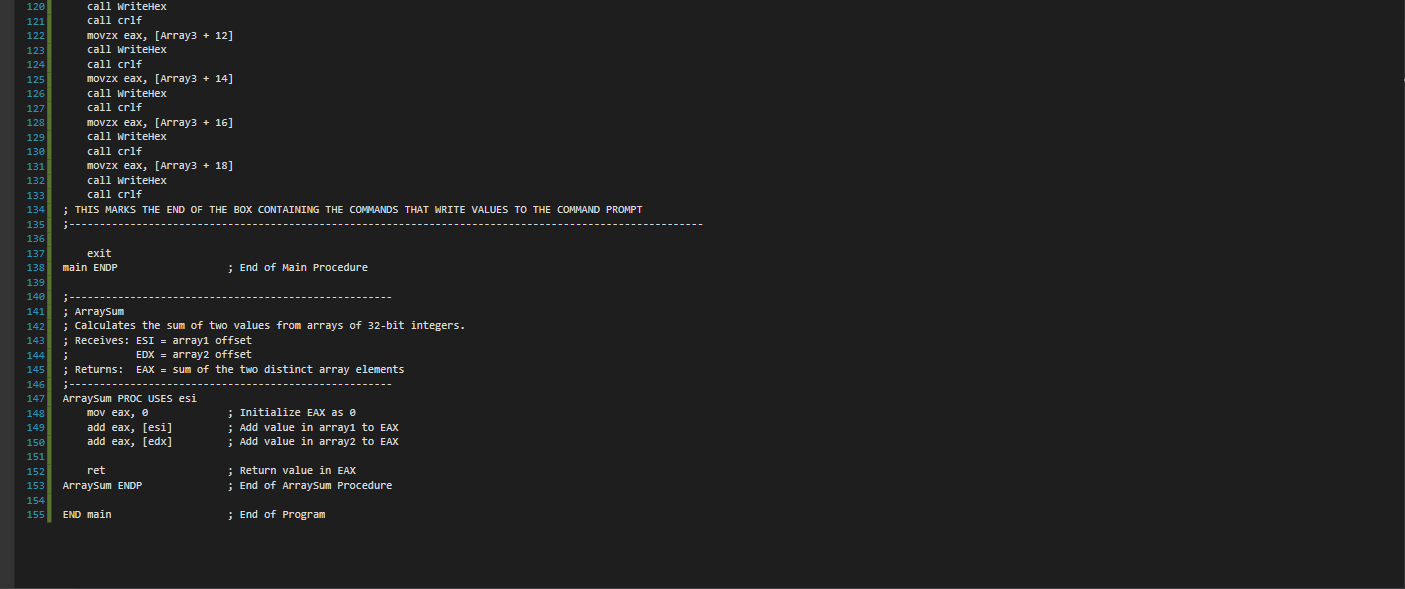
Array3 WORD 10 DUP (?)

**Program Screenshots:**







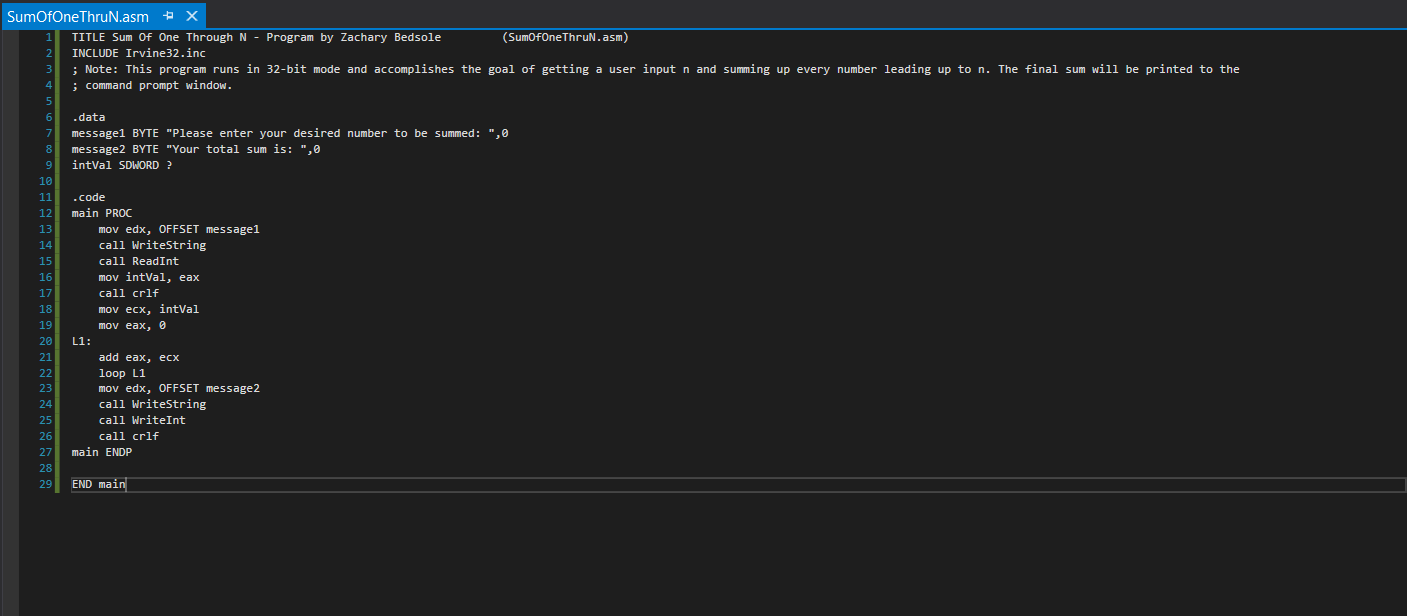


**Output:** 

1. [Loops] Write a program to compute the sum of first *n* integers of the series: *Sum = 1 + 2 + 3 +…* Your program must:
   1. Prompt user for integer *n*,
   2. Read the value of *n* from user input
   3. Calculate *Sum*, and;
   4. Print *Sum* on screen.

Please use the “WriteInt” procedure, not “DumpRegs”. Other relevant procedures: “ReadInt” and “WriteString.” The calculation can be done in many ways, and all submissions that evidence proper programming practice are acceptable. In your homework submission, please embed both the code and one screen shot for *n = 6*.

**Code:**



**Output:**

