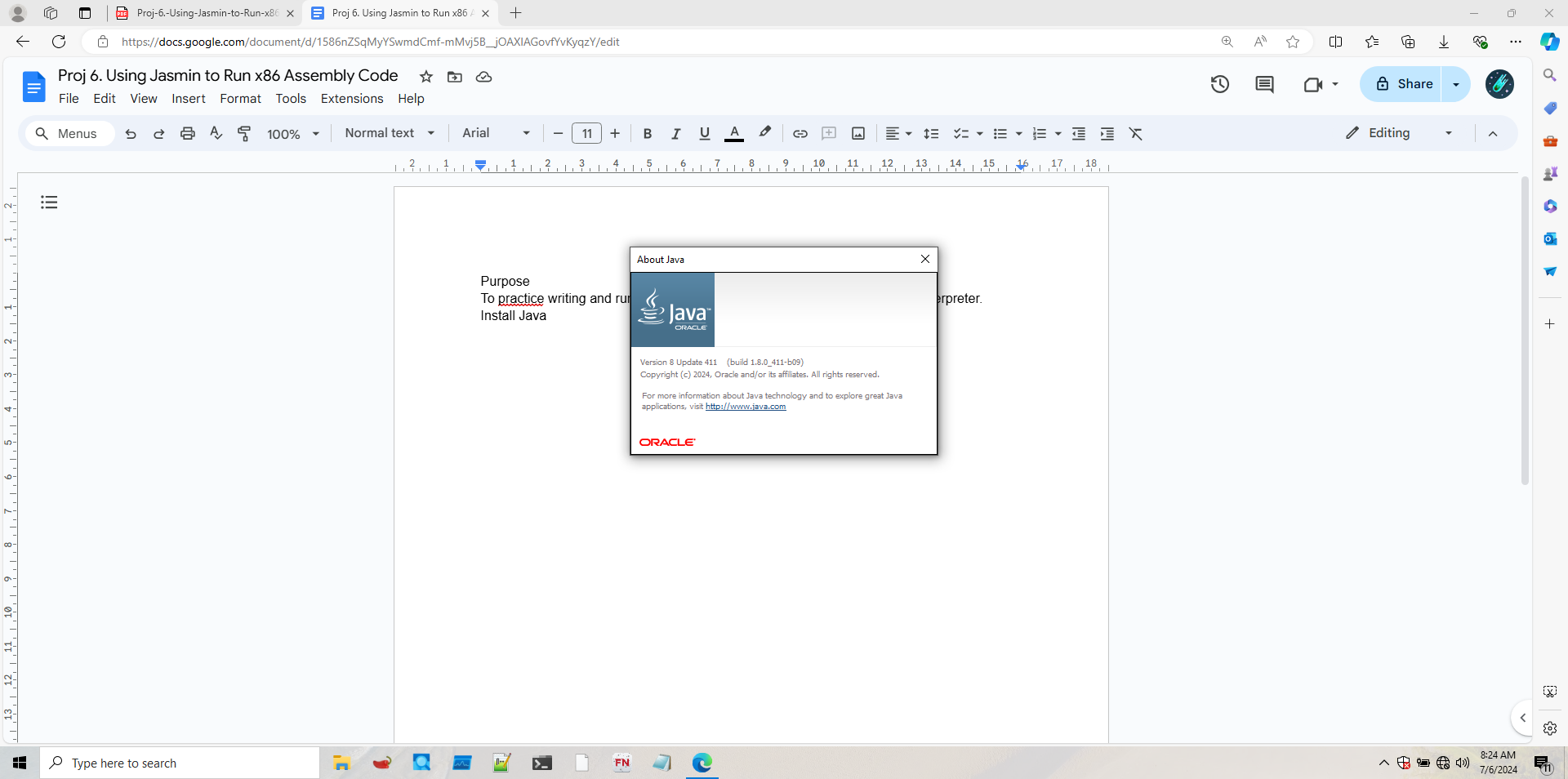
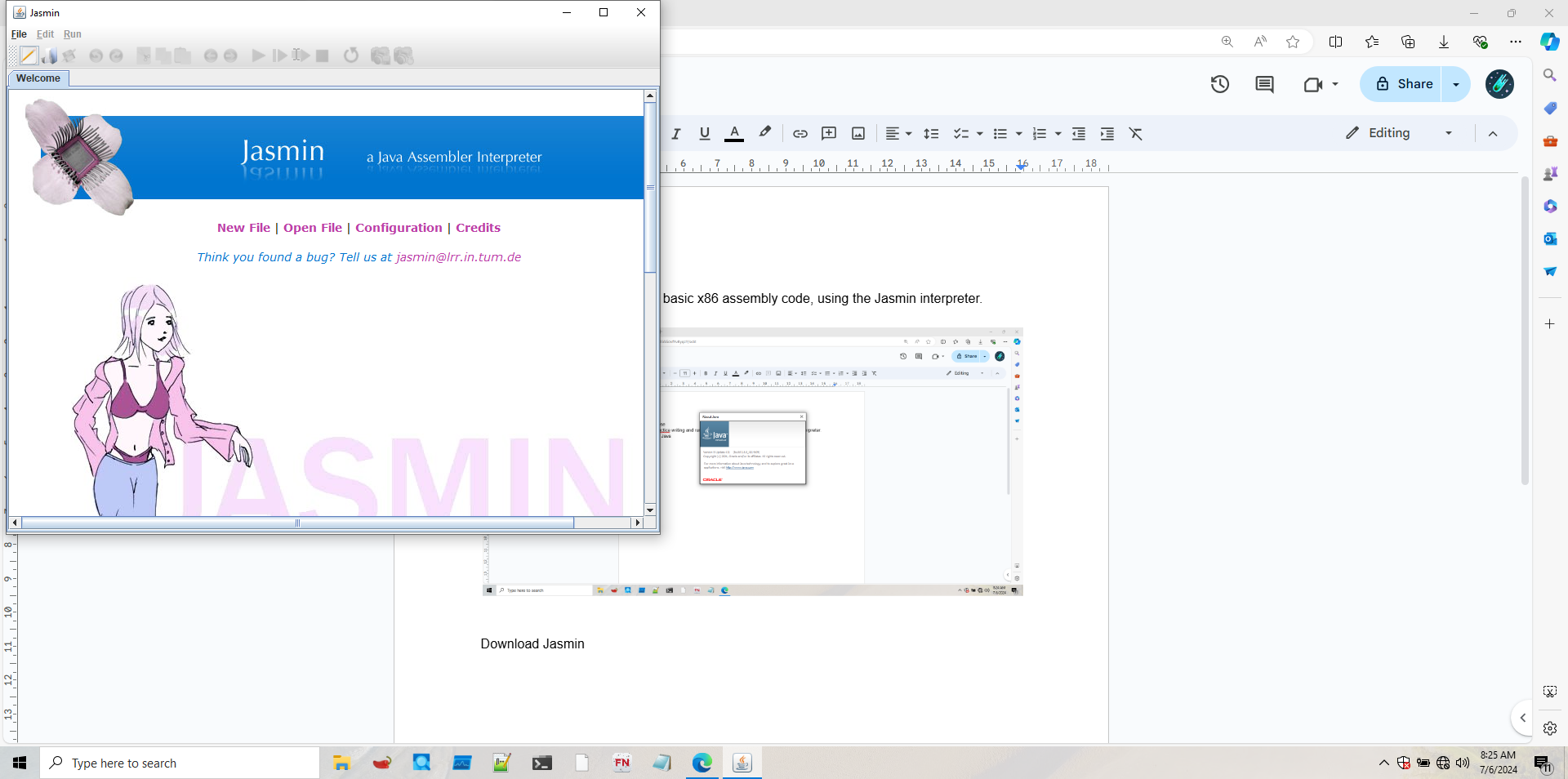
Purpose

To practice writing and running basic x86 assembly code, using the Jasmin interpreter.

Install Java



Download Jasmin

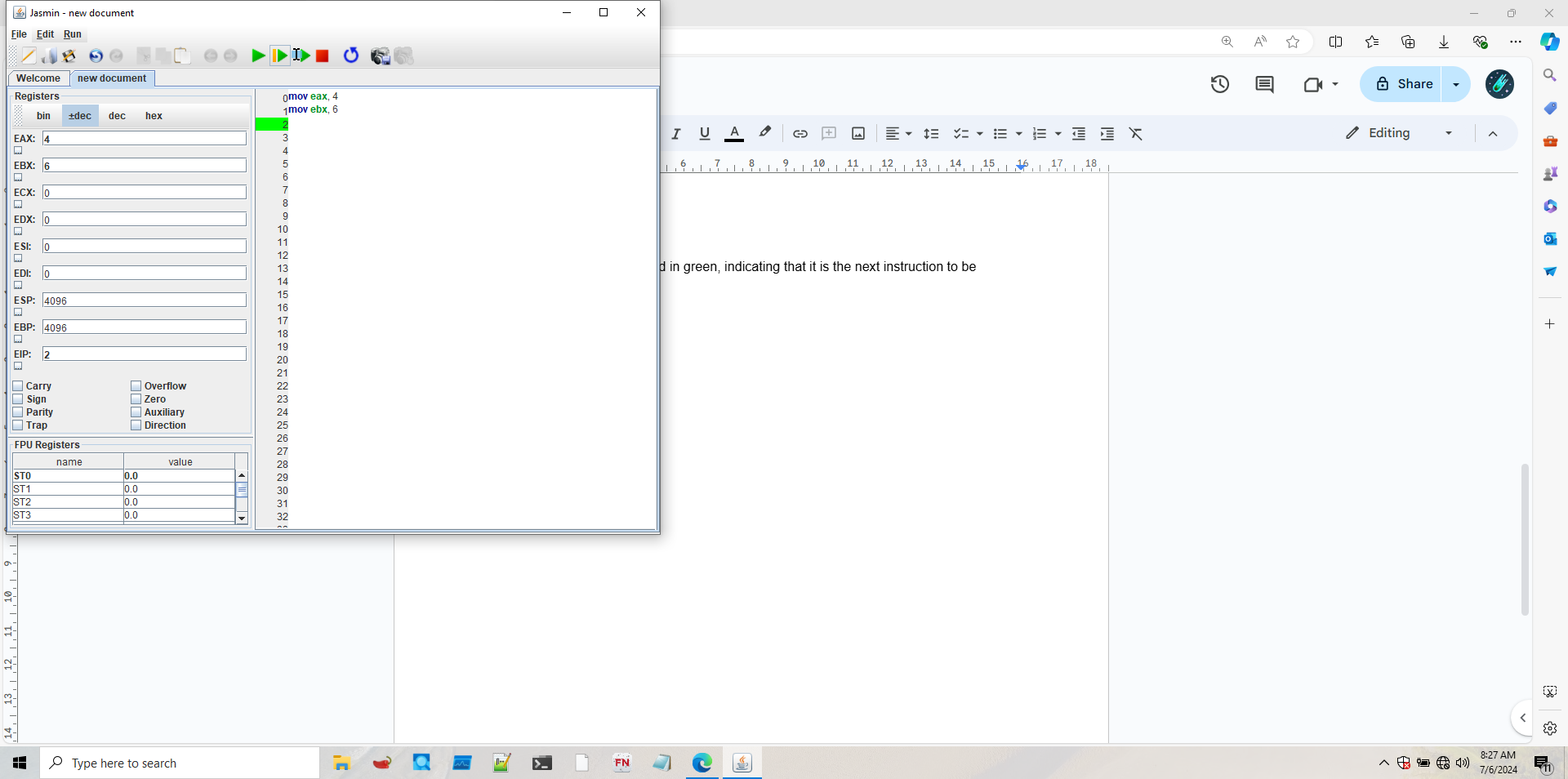


Using mov Instructions In the Code section, type in these instructions.

mov eax, 4

mov ebx, 6

The program runs. When it stops, notice these things, as shown below: EAX contains 4 EBX contains 6 EIP contains 2, because instructions 0 and 1 have been executed In the Code area, instruction 2 is highlighted in green, indicating that it is the next instruction to be processed.



Storing Results in Memory Add more lines to your Code section to make your program look like this

mov eax, 4

mov ebx, 6

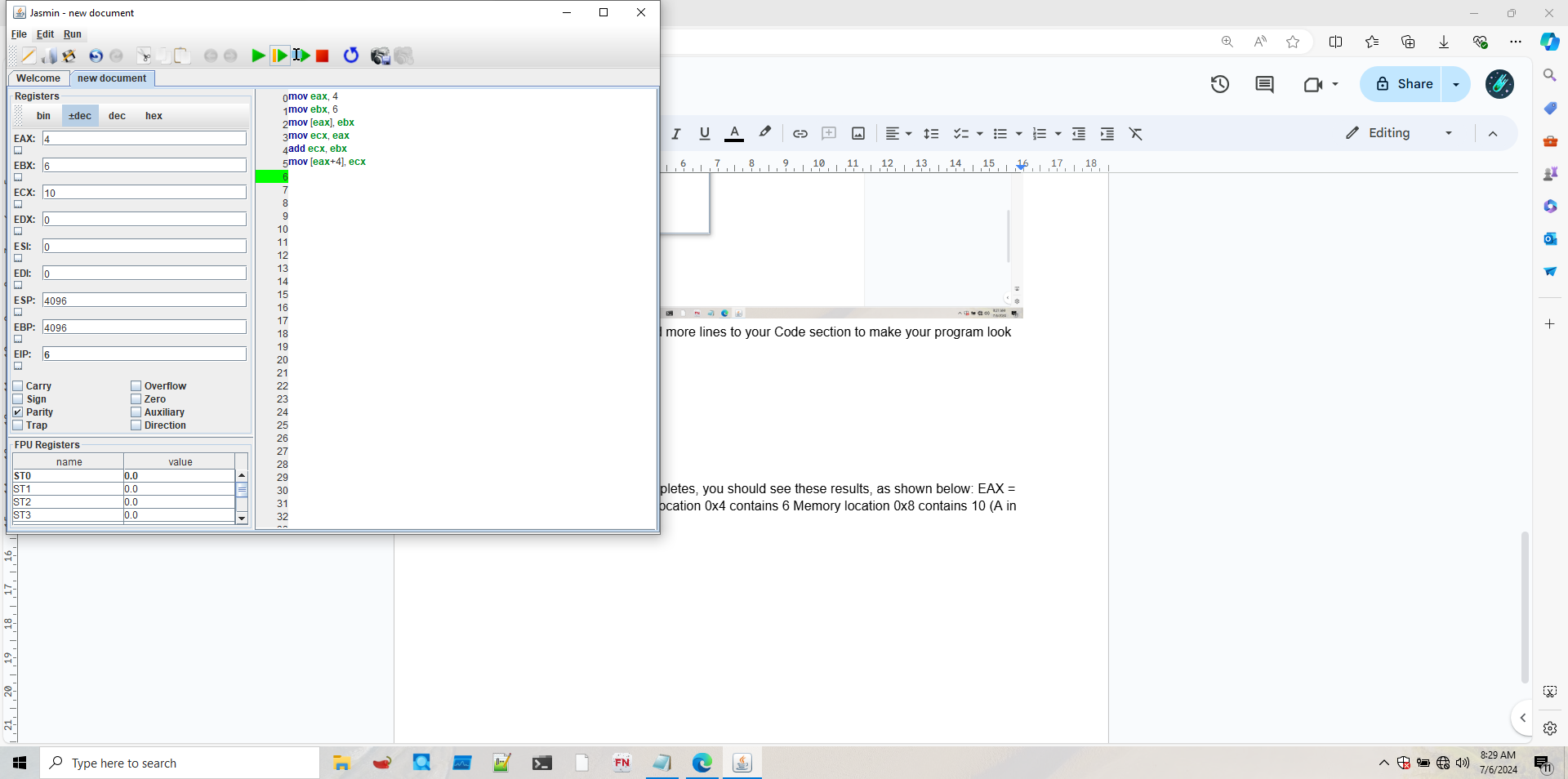
mov [eax], ebx

mov ecx, eax

add ecx, ebx

mov [eax+4], ecx

Run the program. When it completes, you should see these results, as shown below: EAX = 4 EBX = 6 ECX = 10 Memory location 0x4 contains 6 Memory location 0x8 contains 10 (A in hexadecimal)

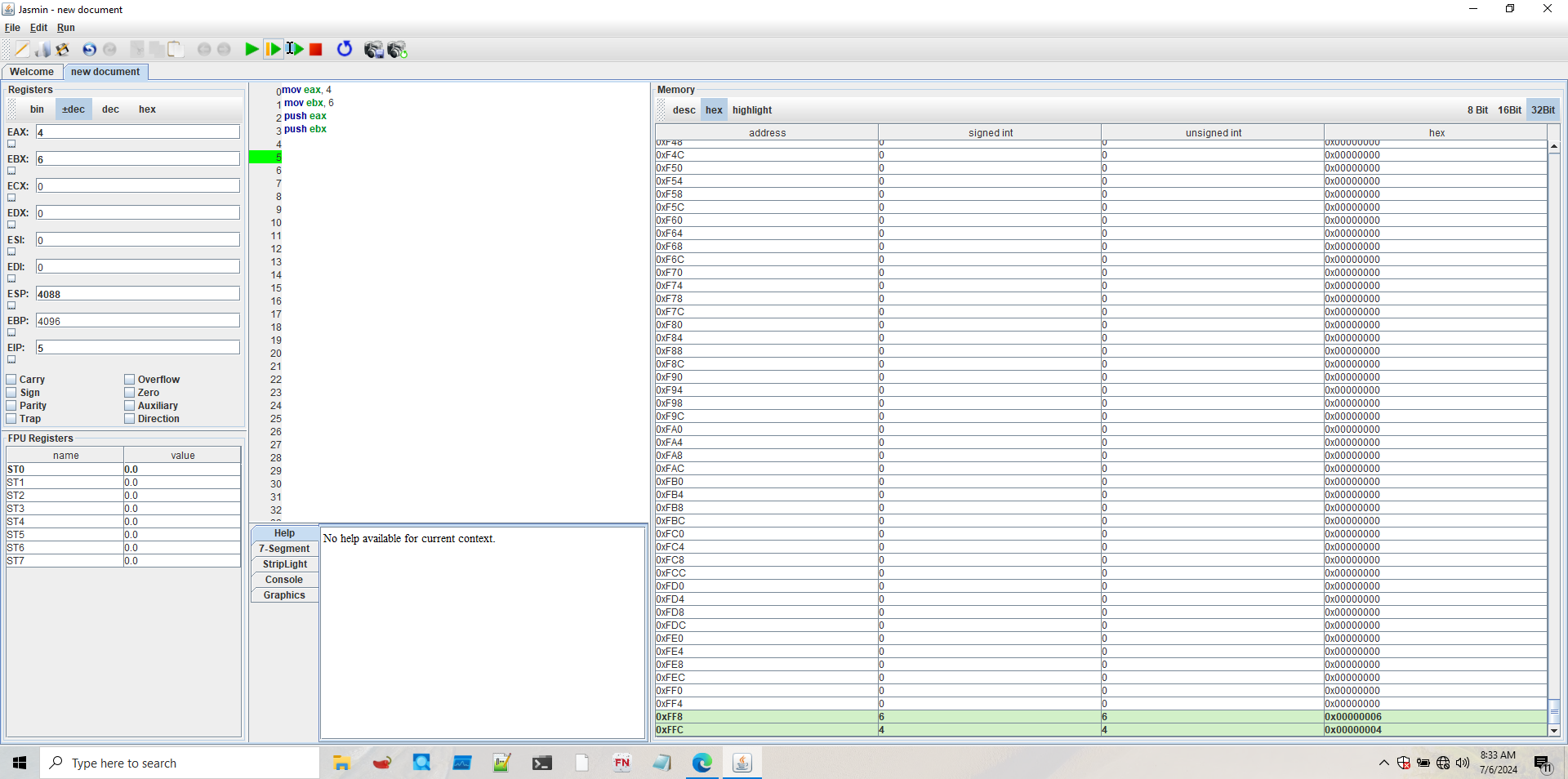


Using the Stack In Jasmin, click File, New. In the Code section, type in these instructions. mov eax, 4

mov ebx, 6

push eax

push ebx



Scroll down in the Memory pane to see the last values. As show below, the last location is at 0xFC. This value is 32 bits long, so it contains four bytes, at locations 0xFC, 0xFD, 0xFE, and 0xFF. The ESP points to the next byte, 0x10

Understanding Push

At the top of the Jasmin window, click the green Run button.

These instructions move the number 4 into eax, and the number 6 into ebx. Then both values are pushed onto the stack.

Notice these things, as shown below:

EAX contains 4

EBX contains 6

ESP contains 248, which is 0xF8, the new top of the stack.

Memory location 0xFC contains 4, the first value pushed onto the stack.

Memory location 0xF8 contains 6, the second value pushed onto the stack

Understanding Pop

Add a pop instruction to your code, so it now looks like this:

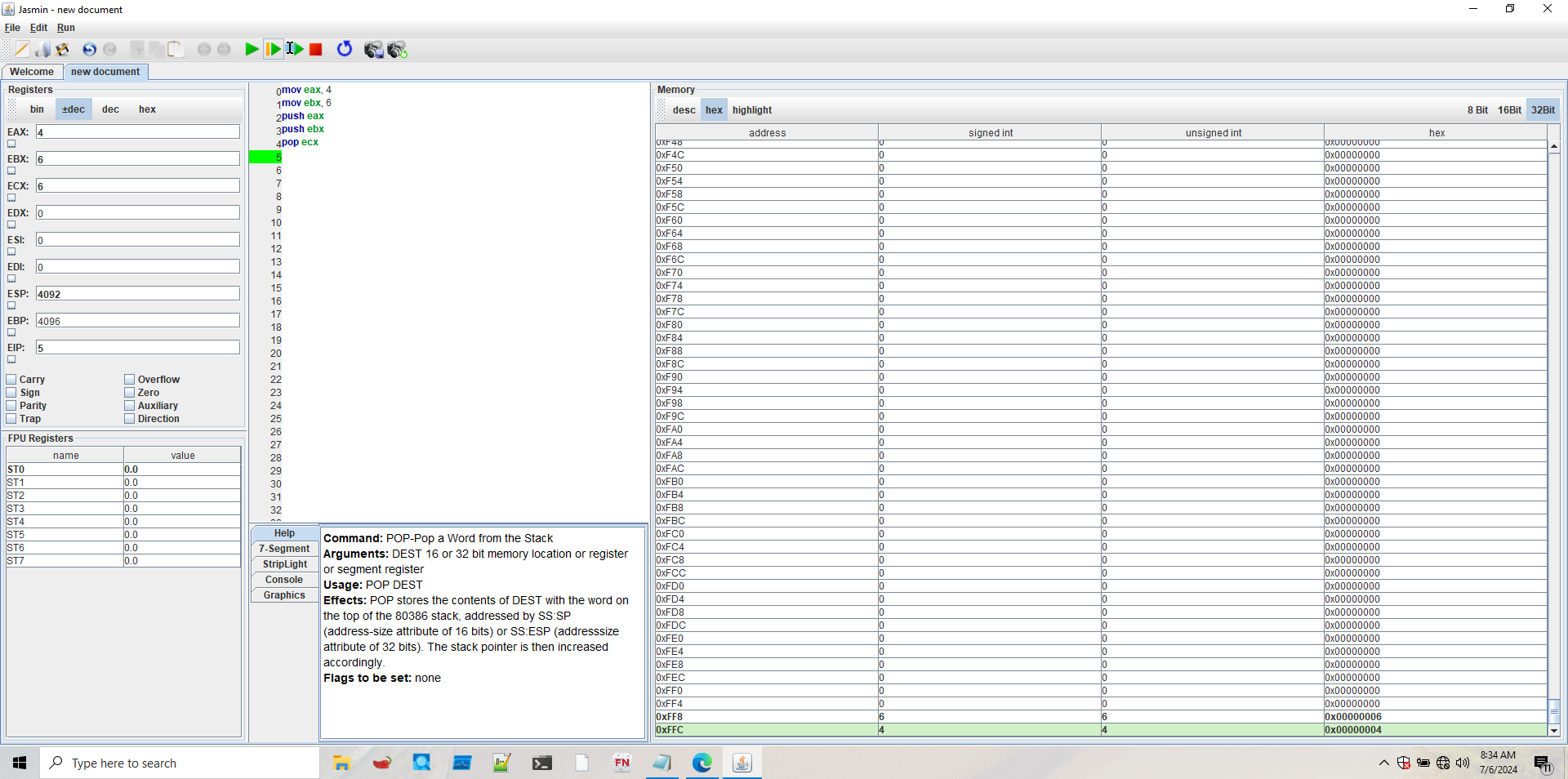
mov eax, 4

mov ebx, 6

push eax

push ebx

pop ecx



ECX contains 6, the value popped off the top of the stack.

ESP contains 252, which is 0xFC, the new top of the stack.

Memory location 0xFC contains 4, the first value pushed onto the stack.

Memory location 0xF8 contains 6, which is now the top value on the stack

Reversing a Sequence

In Jasmin, click File, New.

In the Code section, type in these instructions.

mov eax, 1

mov ebx, 2

mov ecx, 3

mov edx, 4

push eax

push ebx

push ecx

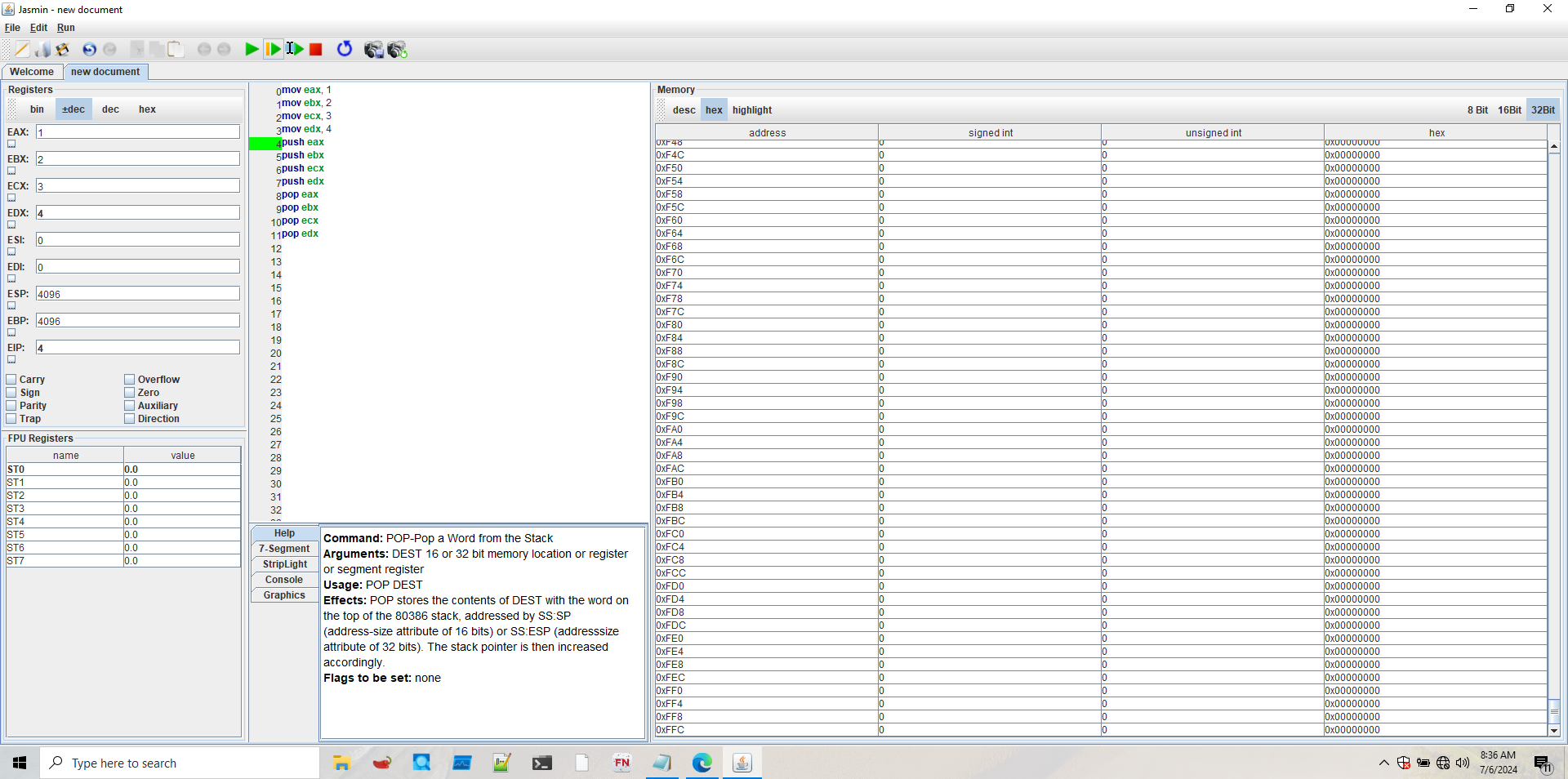
push edx

pop eax

pop ebx

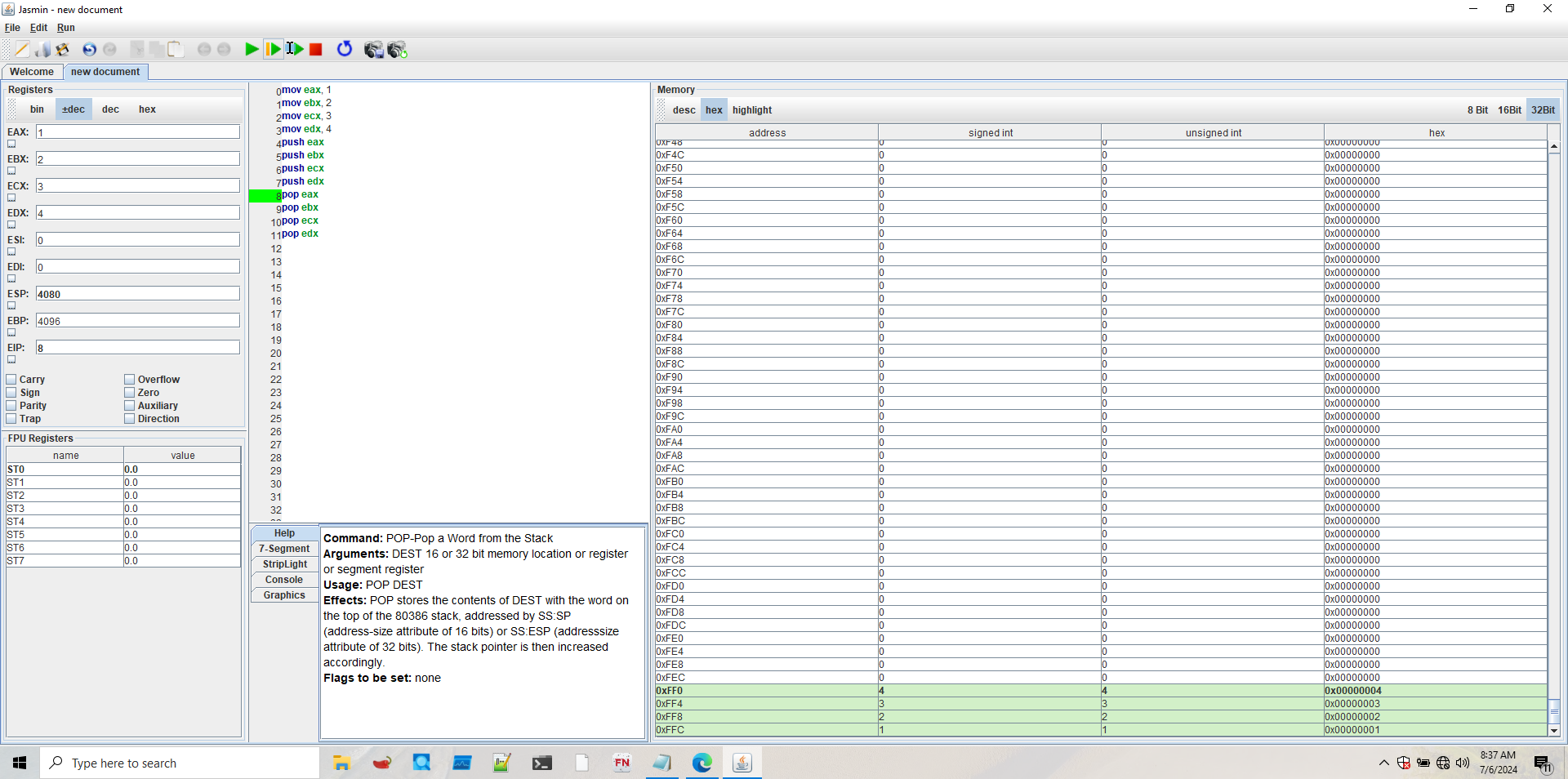
pop ecx

pop ed



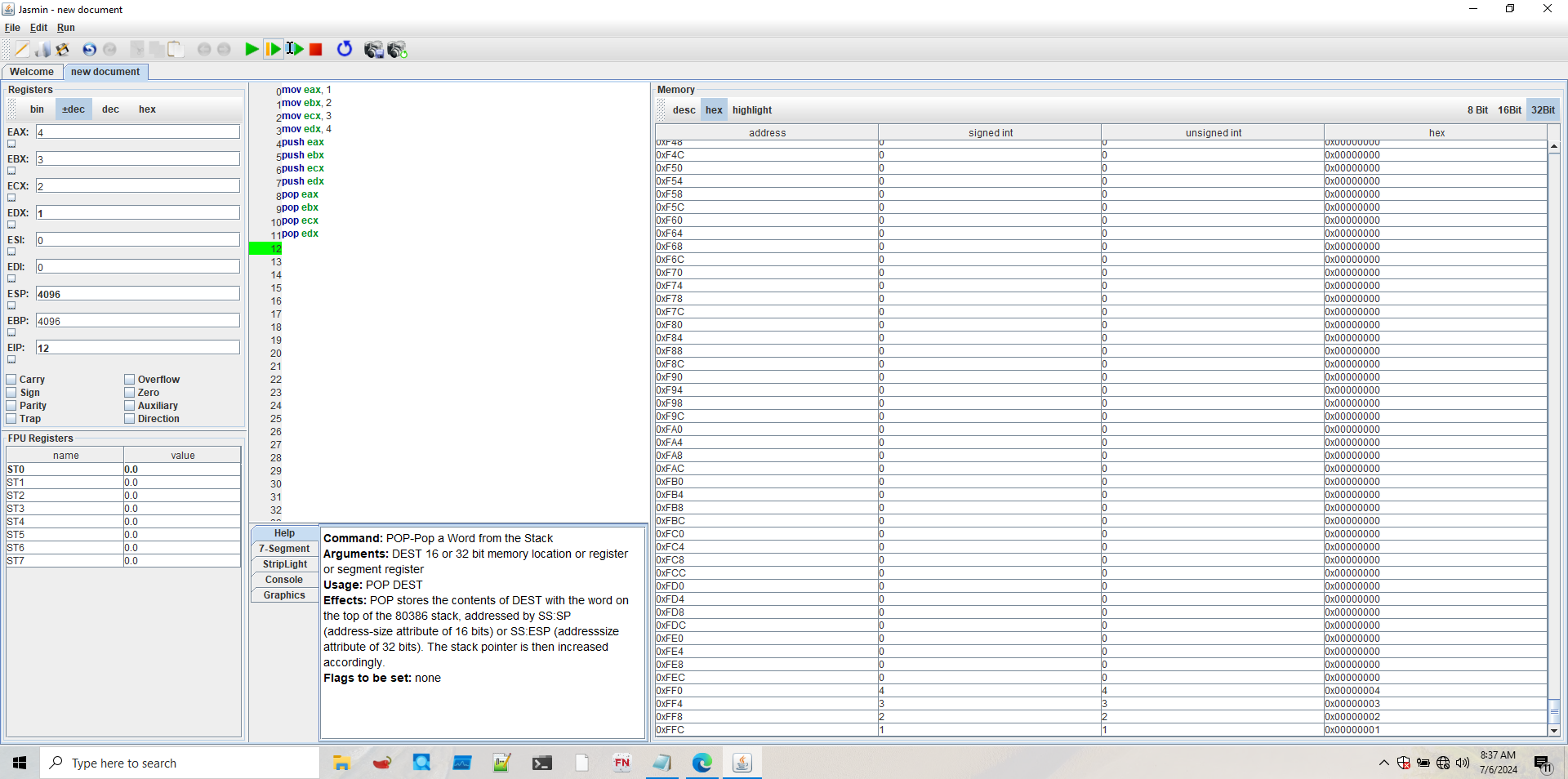
These instructions load values into the four registers, push them onto the stack in order, and pop them off the stack in order. However, since the stack is a FILO (First In, Last Out) structure, this reverses the order of the values. Push the Step four times to execute only the first four instructions, as shown below

Push the Step four more times to execute only the next four instructions. You see the values 1, 2, 3, and 4 pushed onto the stack, as shown below



Push the Step four more times to execute the remaining four instructions.

Now the registers contain these values: EAX = 4 EBX = 3 ECX = 2 EDX = 1



6.1 Find the Value

In Jasmin, click File, New. In the Code section, type in these instructions. mov eax, 1 mov ebx, 9 mov ecx, 49 push eax push ebx push eax push ecx push ebx push eax pop eax pop ebx pop ecx add eax, ebx pop ecx add eax, ecx pop ebx pop ecx add eax, ebx Load and run this program. Find the value of eax

