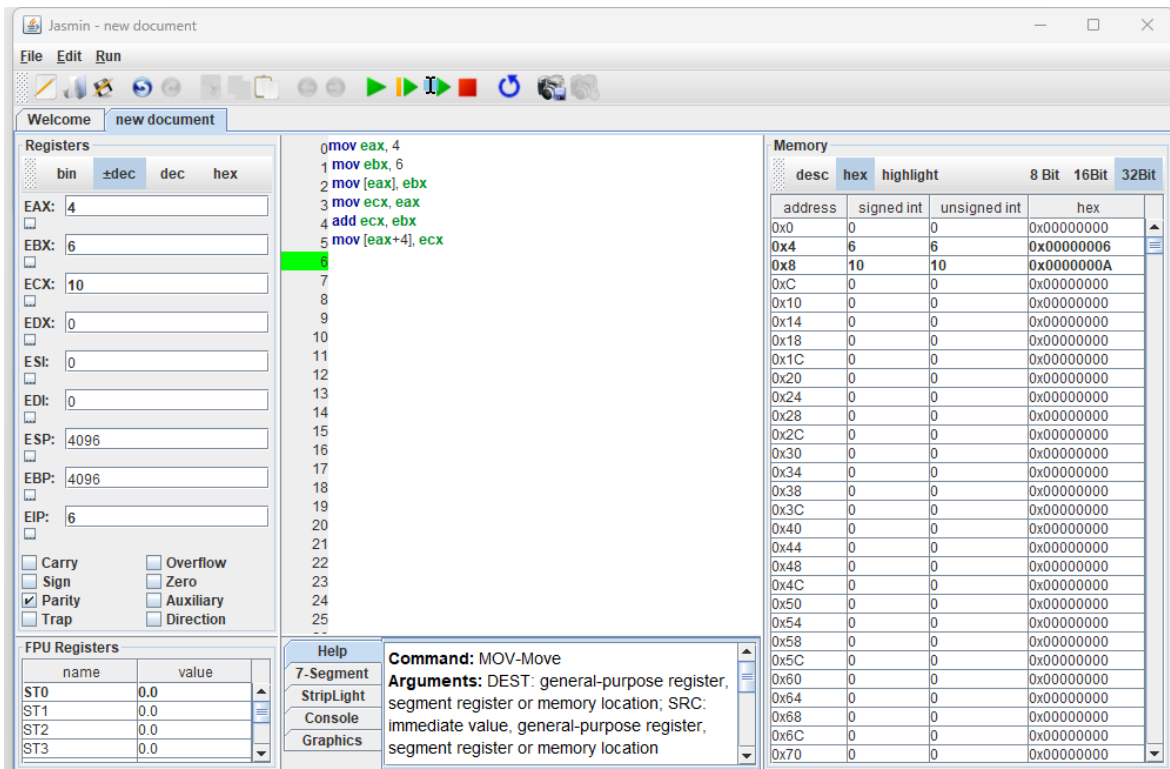


LAB 9: Using Jasmin to run x86 Assembly Code

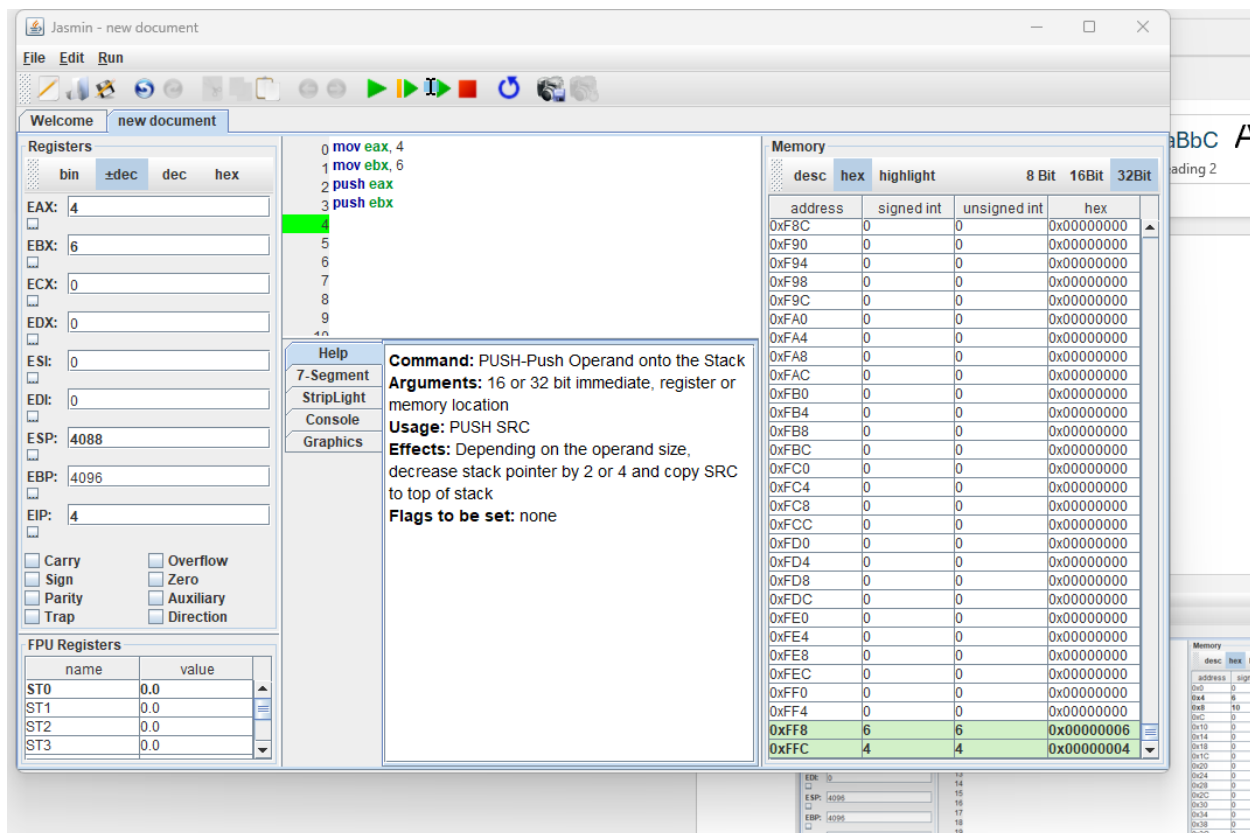
Storing Results in Memory

- EAX = 4
- EBX = 6
- ECX = 10
- Memory location 0x4 contains 6
- Memory location 0x8 contains 10



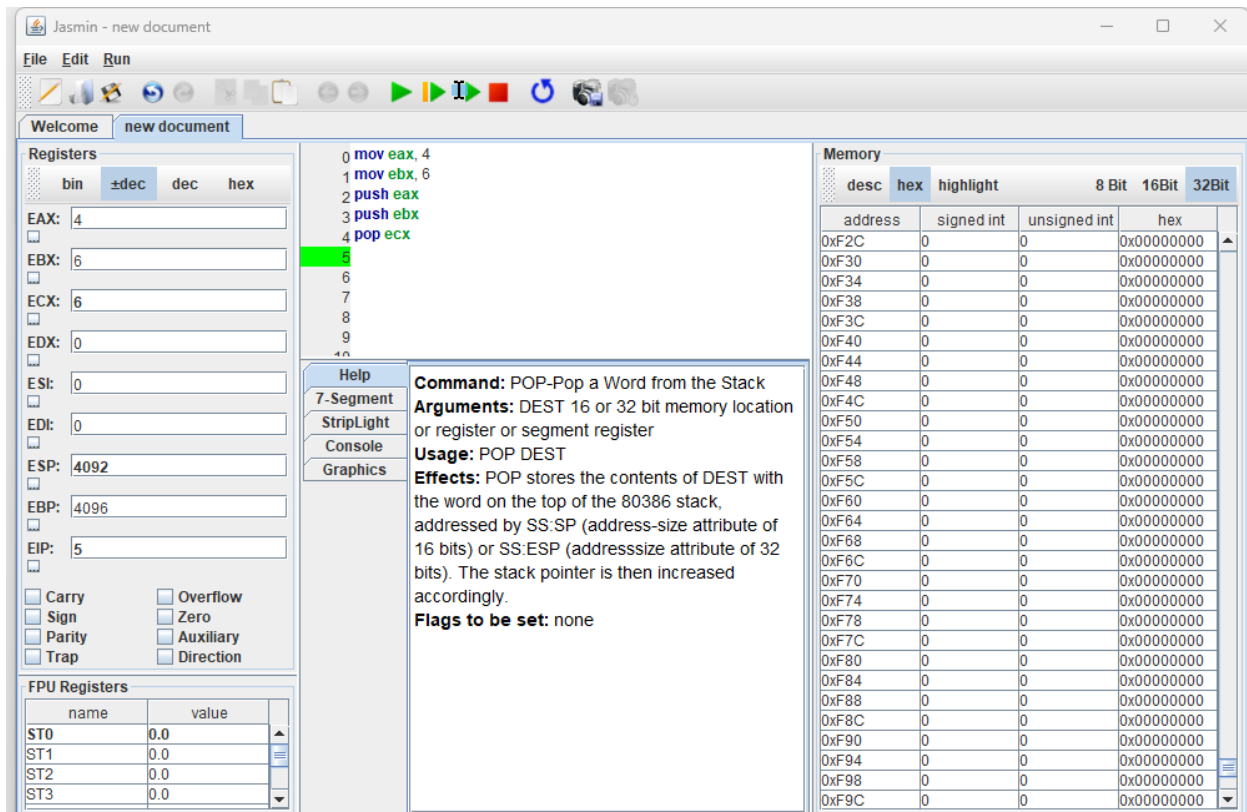
Understanding Push

- EAX contains 4
- EBX contains 6
- ESP contains 4088, which is 0xFF8, the new top of the stack.
- Memory location 0xFFC contains 4, the first value pushed onto the stack.
- Memory location 0xFF8 contains 6, the second value pushed onto the stack.



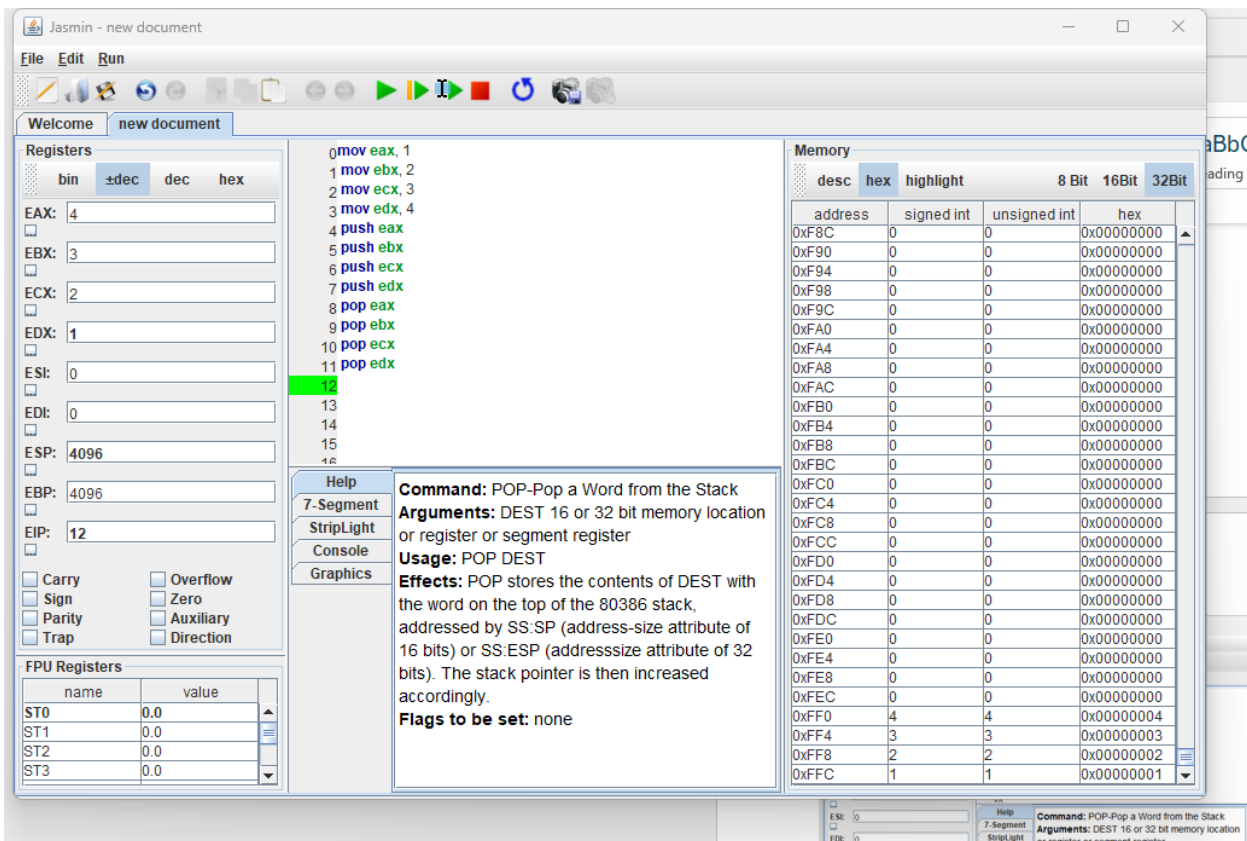
Understanding Pop

- ECX contains 6, the value popped off the top of the stack.
- ESP contains 4092, which is 0xFFC, the new top of the stack.
- Memory location 0xFFC contains 4, the first value pushed onto the stack.
- Memory location 0xFF8 contains 6, which is now the top value on the stack.

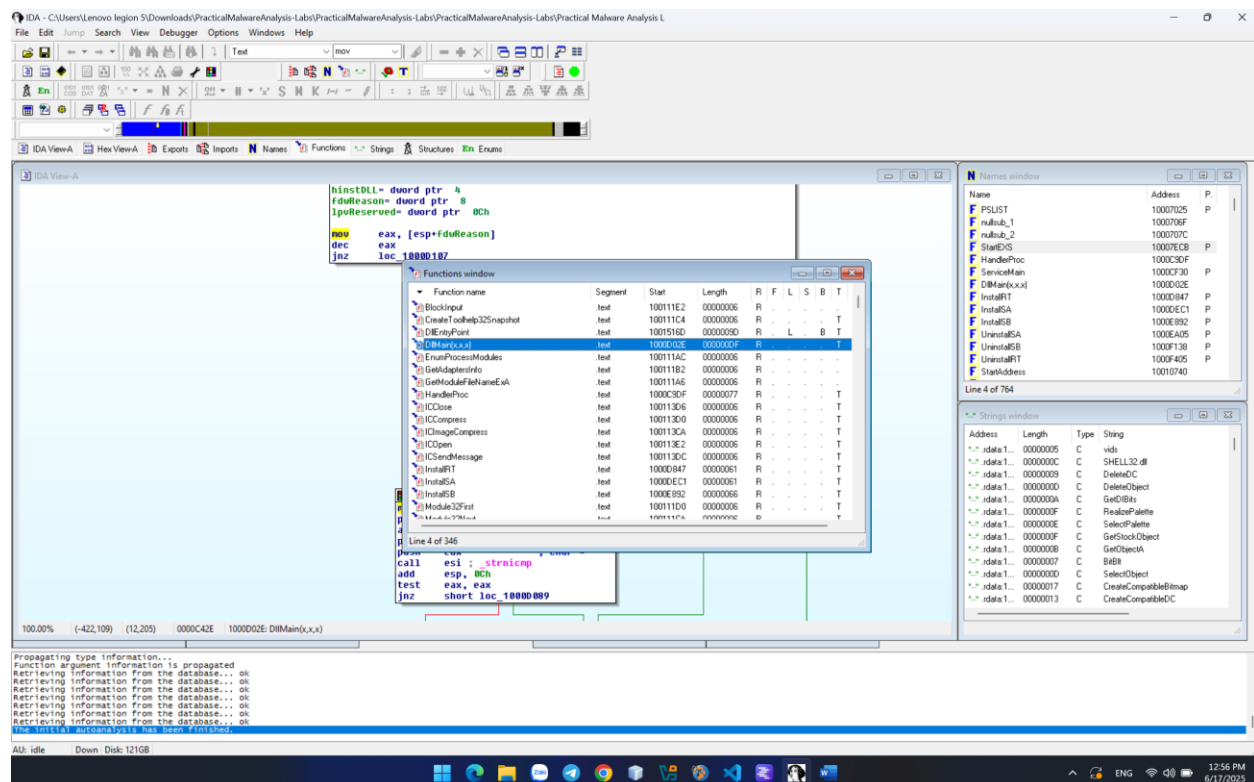


Reversing a Sequence

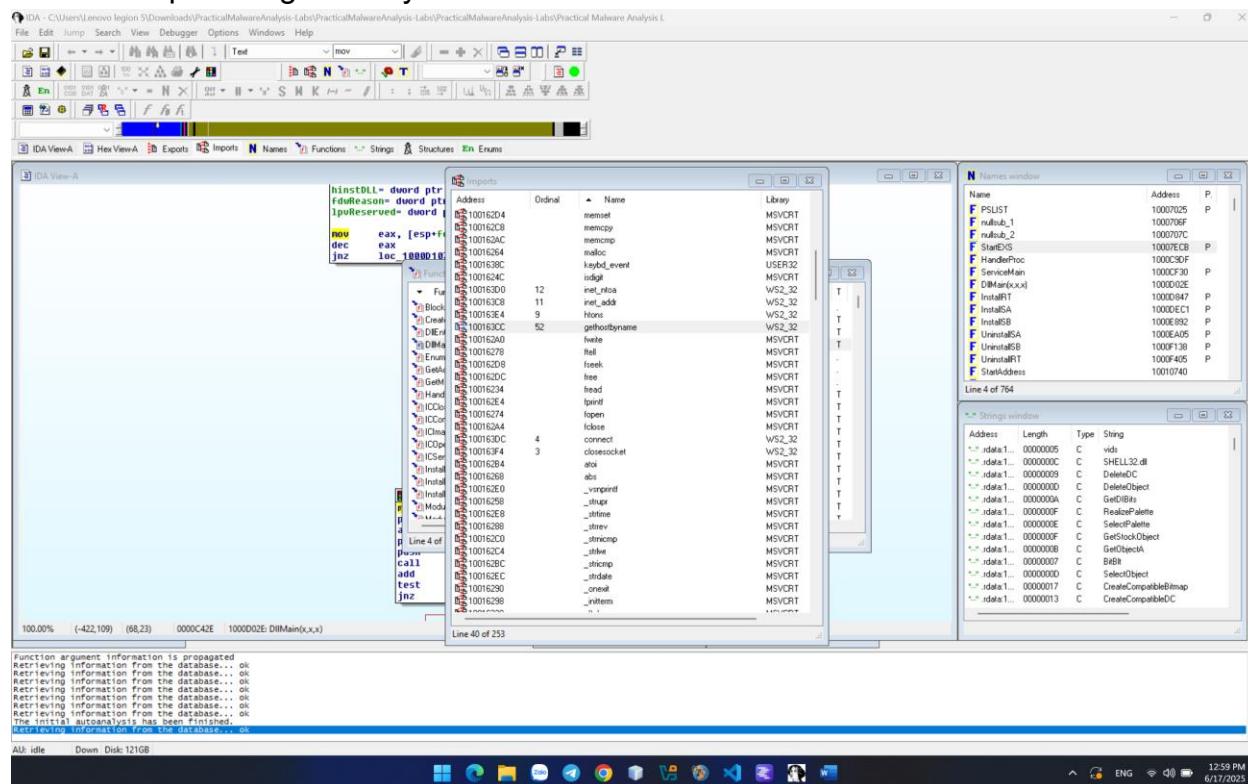
EAX = 4 EBX = 3 ECX = 2 EDX = 1



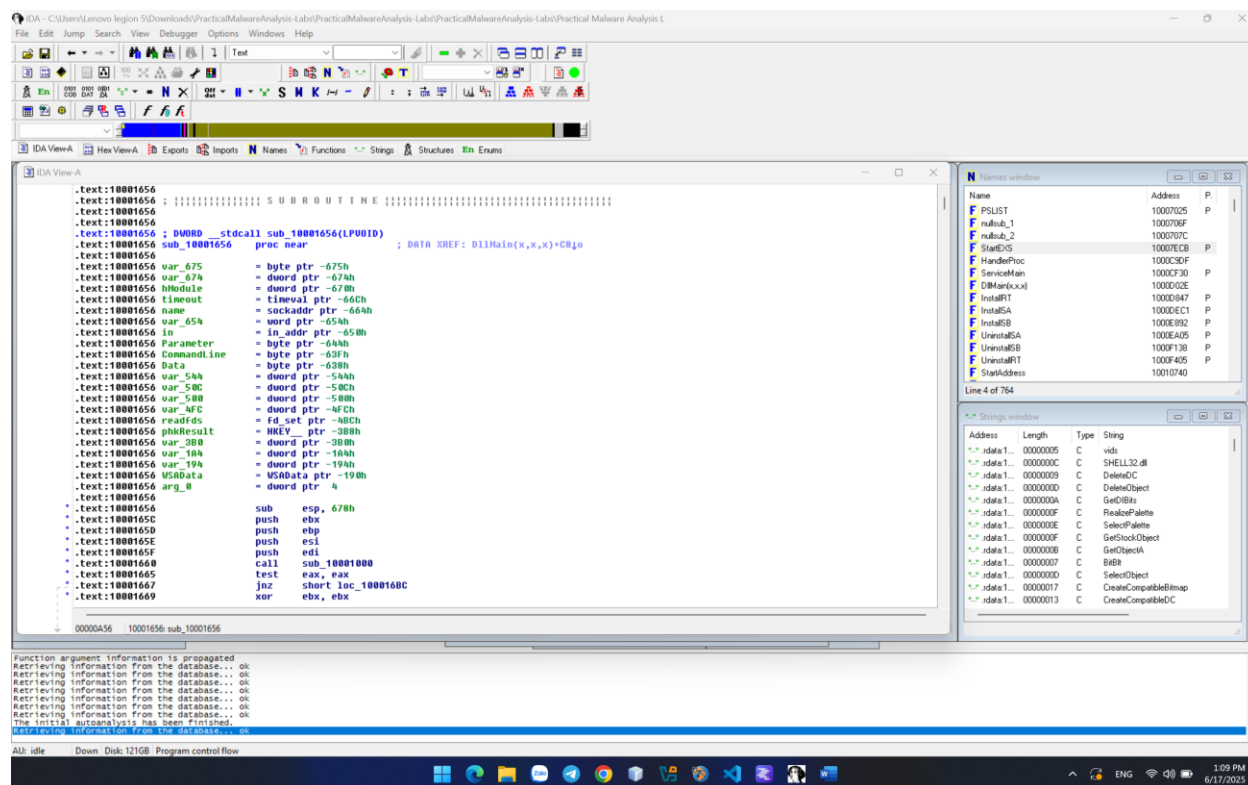
LAB 9.1: Lab05-01.dll in IDA Pro



Find the import for gethostbyname

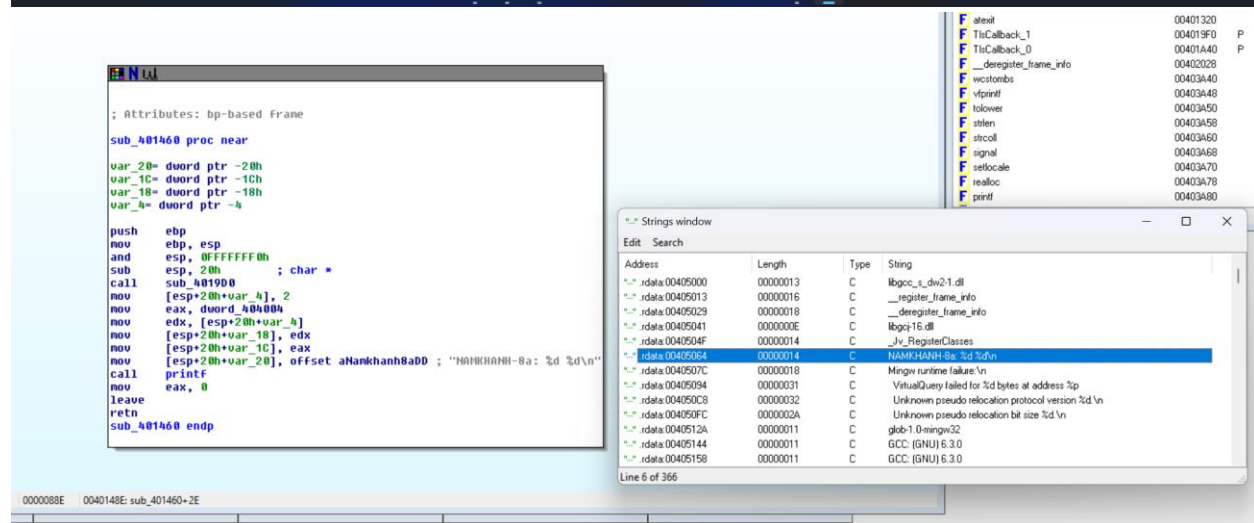
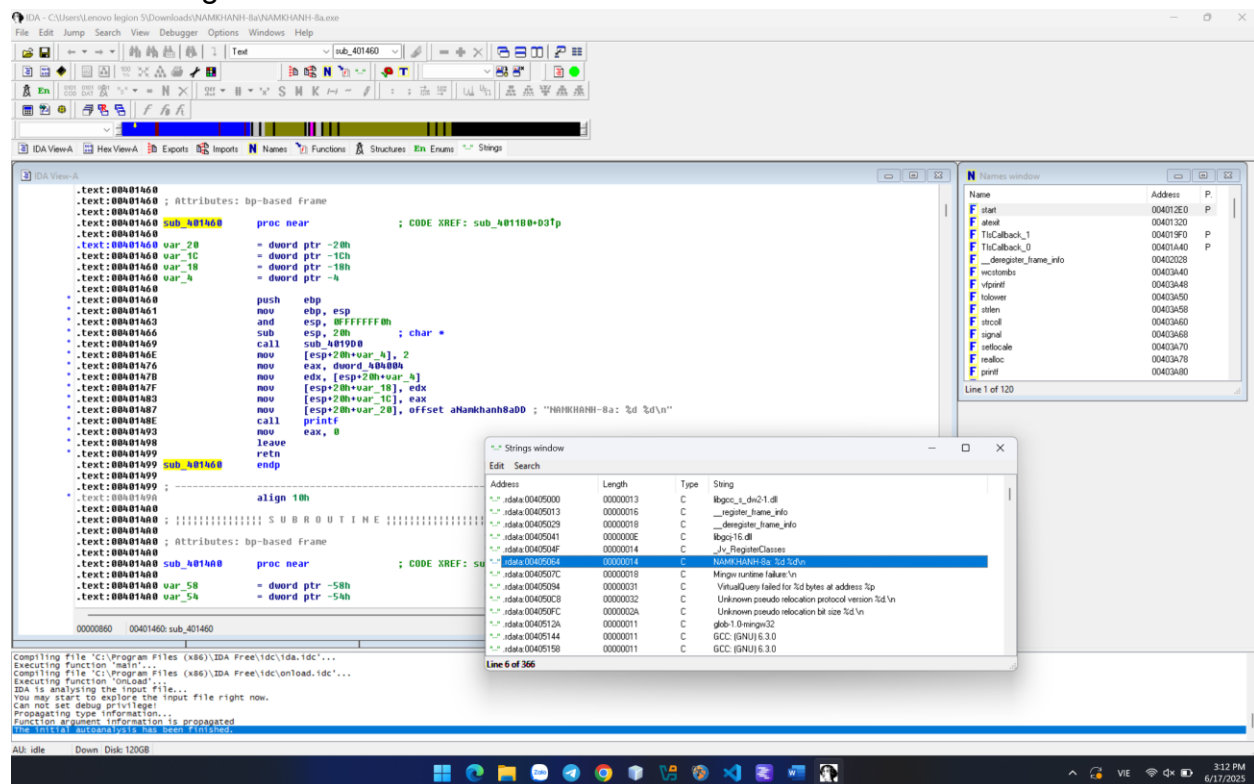


Count Local Variables for the Subroutine at 0x10001656



[illegible]

Disassembling the EXE



CHALLENGE:

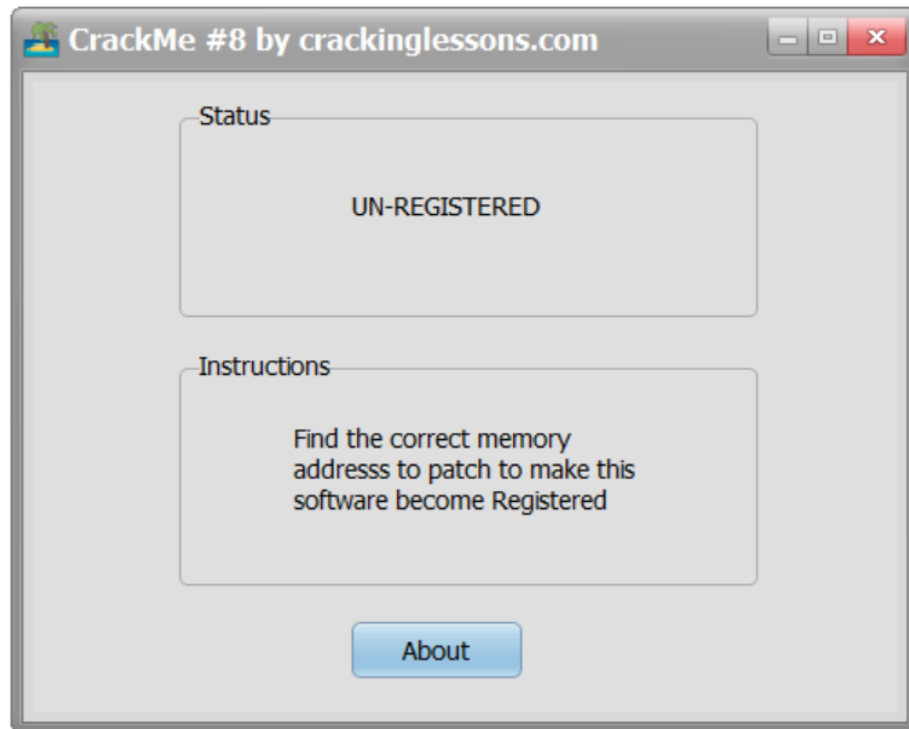
Code:

```
13
14  #include <stdio.h>
15
16  int x = 10;    // Biến toàn cục x
17  int m = 20;    // Biến toàn cục m
18
19  int main() {
20      int y = 30;    // Biến cục bộ y
21      int n = 40;    // Biến cục bộ thứ hai
22
23      printf("NAMKHANH %d %d %d %d", x, m, y, n); // Thay "NAMKHANH" bằng tên của bạn
24      return 0;
25 }
```

Disassembling the EXE

CRACKME 8:

This crackme is for learning how to put hardware breakpoints on memory addresses and then patch it to register the program.



Extract the .exe file, run it for testing, and note down significant keywords. In this situation: UN-REGISTERED.

Then load the file into x32dbg, open the String references section to locate that keyword. Click on the keyword to find the corresponding code line.

```
mov edx,dword ptr ds:[725560]      00725560 "<an>"
mov ecx,dword ptr ds:[725568]      00725568 "Han"
mov esi,dword ptr ds:[725564]      00725564 "Dan"
mov dword ptr ss:[esp+4],crackme8.718154 00718154 "Slate Classico"
mov dword ptr ss:[esp+4],crackme8.718163 00718163 "CrackMe #8"
mov dword ptr ss:[esp+4],crackme8.7181EE 007181EE "REGISTERED"
mov dword ptr ss:[esp+4],crackme8.7181E0 007181E0 "UN-REGISTERED"
mov edx,crackme8.7181FA            007181FA L"coded by crackinglessons.com"
mov ecx,crackme8.718234            00718234 L>About"
add byte ptr ds:[ecx+ebx+76080057],cI    76080057 "ram cannot be run in DOS mode.\r\r\n$!"
push 17261                        00017261 "!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!"
imul ebp,dword ptr ds:[esi+67],20000   00020000 "!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!"
imul esp,dword ptr ss:[ebp+73],20002    00020002 "!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!"
imul esi,dword ptr ds:[eax+eax],crackme8.403E5C 0A03E5C ">&"
imul esi,dword ptr ds:[eax+eax],crackme8.403E5C 0A03E5C ">&"
imul ebp,dword ptr ds:[esi+67],crackme8.404040 0A04040 &"\nAnsiString"
imul edi,dword ptr ds:[edx+65],20002     00020002 "!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!"
imul edi,dword ptr ds:[edx+65],20002     90020002 "!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!"
add byte ptr ds:[ecx+20040],cI          00020040 "!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!"
add byte ptr ds:[ecx+20040],cI          00020040 "!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!"
cbt byte ptr ds:[ecx+20040],aI         00020040 "!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!"
cmp edi,10A60                       00010A60 "!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!"
mov eax,dword ptr ds:[718358]           00718358 "&\r\n"
mov eax,dword ptr ds:[718358]           00718358 "&\r\n"
mov eax,dword ptr ds:[71834C]           0071834C "&'Unknown'"
mov eax,dword ptr ds:[71834C]           0071834C "&'Unknown'"
```

Pay attention to the code section above "UN-REGISTERED".

00403855	E8 62752600	call crackme8.66ADBC	
0040385A	833D 00607200 00	cmp dword ptr ds:[7260D0],0	
00403861	8B45 EC	mov eax,dword ptr ss:[ebp-14]	
00403864	8B80 D4030000	mov eax,dword ptr ds:[eax+3D4]	
0040386A	8945 DC	mov dword ptr ss:[ebp-24],eax	
0040386D	74 38	je crackme8.4038A7	
0040386F	C745 E4 00000000	mov dword ptr ss:[ebp-1C],0	
00403876	C745 AC 07000000	mov dword ptr ss:[ebp-54],7	
0040387D	8D45 E4	lea eax,dword ptr ss:[ebp-1C]	
00403880	890424	mov dword ptr ss:[esp],eax	
00403883	C74424 04 EE817100	mov dword ptr ss:[esp+4],crackme8.7181EE	7181EE:"REGISTERED"
00403886	E8 34293100	call crackme8.7161C4	
00403890	8B55 E4	mov edx,dword ptr ss:[ebp-1C]	
00403893	8B45 DC	mov eax,dword ptr ss:[ebp-24]	
00403896	C745 AC 08000000	mov dword ptr ss:[ebp-54],8	
0040389D	E8 9E501600	call crackme8.568940	
004038A2	8D45 E4	lea eax,dword ptr ss:[ebp-1C]	
004038A5	EB 36	jmp crackme8.4038D0	
004038A7	C745 E0 00000000	mov dword ptr ss:[ebp-20],0	
004038AE	C745 AC 05000000	mov dword ptr ss:[ebp-54],5	
004038B5	8D45 E0	lea eax,dword ptr ss:[ebp-20]	
004038B8	890424	mov dword ptr ss:[esp],eax	
004038BB	C74424 04 E0817100	mov dword ptr ss:[esp+4],crackme8.7181E0	7181E0:"UN-REGISTERED"
004038C3	E8 FC283100	call crackme8.7161C4	
004038C8	8B55 E0	mov edx,dword ptr ss:[ebp-20]	
004038CB	8B45 DC	mov eax,dword ptr ss:[ebp-24]	

Locate the processing section that causes the program to display the "UN-REGISTERED" status. This is the code section shown below:

0040386D	74 38	je crackme8.4038A7	
0040386F	C745 E4 00000000	mov dword ptr ss:[ebp-1C],0	
00403876	C745 AC 07000000	mov dword ptr ss:[ebp-54],7	
0040387D	8D45 E4	lea eax,dword ptr ss:[ebp-1C]	
00403880	890424	mov dword ptr ss:[esp],eax	
00403883	C74424 04 EE817100	mov dword ptr ss:[esp+4],crackme8.7181EE	7181EE:"REGISTERED"
00403886	E8 34293100	call crackme8.7161C4	
00403890	8B55 E4	mov edx,dword ptr ss:[ebp-1C]	
00403893	8B45 DC	mov eax,dword ptr ss:[ebp-24]	
00403896	C745 AC 08000000	mov dword ptr ss:[ebp-54],8	
0040389D	E8 9E501600	call crackme8.568940	
004038A2	8D45 E4	lea eax,dword ptr ss:[ebp-1C]	
004038A5	EB 36	jmp crackme8.4038D0	
004038A7	C745 E0 00000000	mov dword ptr ss:[ebp-20],0	
004038AE	C745 AC 05000000	mov dword ptr ss:[ebp-54],5	
004038B5	8D45 E0	lea eax,dword ptr ss:[ebp-20]	
004038B8	890424	mov dword ptr ss:[esp],eax	
004038BB	C74424 04 E0817100	mov dword ptr ss:[esp+4],crackme8.7181E0	7181E0:"UN-REGISTERED"
004038C3	E8 FC283100	call crackme8.7161C4	

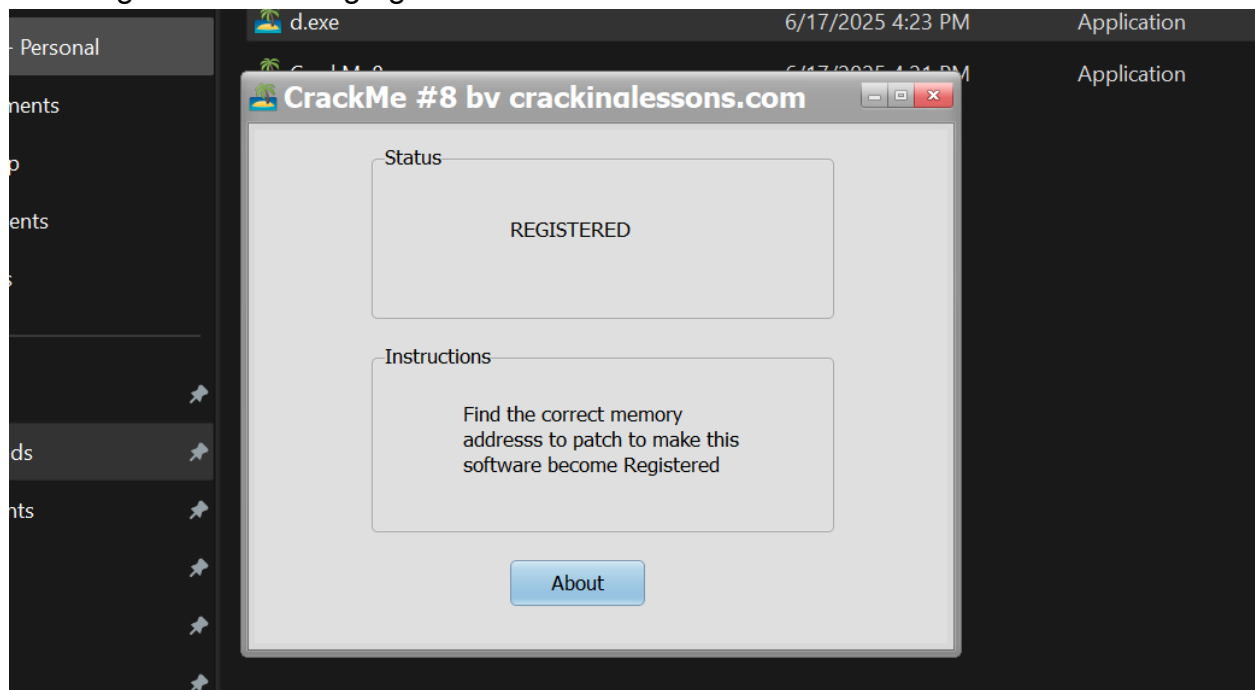
The JE instruction will jump to the "UN-REGISTERED" status processing section if the condition of the preceding CMP instruction is satisfied. Below are the details of the CMP instruction:

E8 62752600	call crackme8.66ADBC	
833D 00607200 00	cmp dword ptr ds:[7260D0],0	
8B45 EC	mov eax,dword ptr ss:[ebp-14]	
8B80 D4030000	mov eax,dword ptr ds:[eax+3D4]	
8945 DC	mov dword ptr ss:[ebp-24],eax	
74 38	je crackme8.4038A7	

Compare the PTR DS variable with the value 0. If the condition is met, the JZ flag will be set and the JE instruction will be executed. To make the program display "REGISTERED" status instead of "UN-REGISTERED", we need to change the comparison value with the PTR DS variable to a non-zero value. Let's try changing it to 1.

00403855	E8 62752600	call crackme8.66ADBC	
0040385A	833D 00607200 01	cmp dword ptr ds:[7260D0],1	
00403861	8B45 EC	mov eax,dword ptr ss:[ebp-14]	
00403864	8B80 D4030000	mov eax,dword ptr ds:[eax+3D4]	

Patching file and running again:



DONE!