

Make SURE that you set your DOS window's properties to black-on-white

1. Create a word document named **Quiz11Snapshots** with your name, quiz number, and date across the top. You are going to paste snapshots into the document.

Download the zipped folder quiz10.zip from D2L content. That folder contains the file quiz10.asm, and the file quiz10.exe. You won't need any other files. Use windbg and step through the program.

Task1: Once you have started the debugger, view the registers and take a snapshot of your registers. You may also want to write down the contents of the stack pointer because you will need it later.

Task 2 – Read CAREFULLY the following instructions

IMPORTANT: use the "Step into" button for EVERY instruction EXCEPT call **getcharecho**. When you get ready to execute that instruction use the "Step OVER" because you do not have the source code for that routine. If you mistakenly hit the Step Into, you will have to start over.

When you hit the step over button for the getcharecho, it will seem that the program freezes..that's because the program is waiting for you to input a character into windbg's DOS window. You will know the correct window for input because it will be white-on-black. Once you type a character, you will need to go back to the source file – the yellow line will have moved down to the next line.

When requested for input, type the characters one at a time when requested: ASHPT

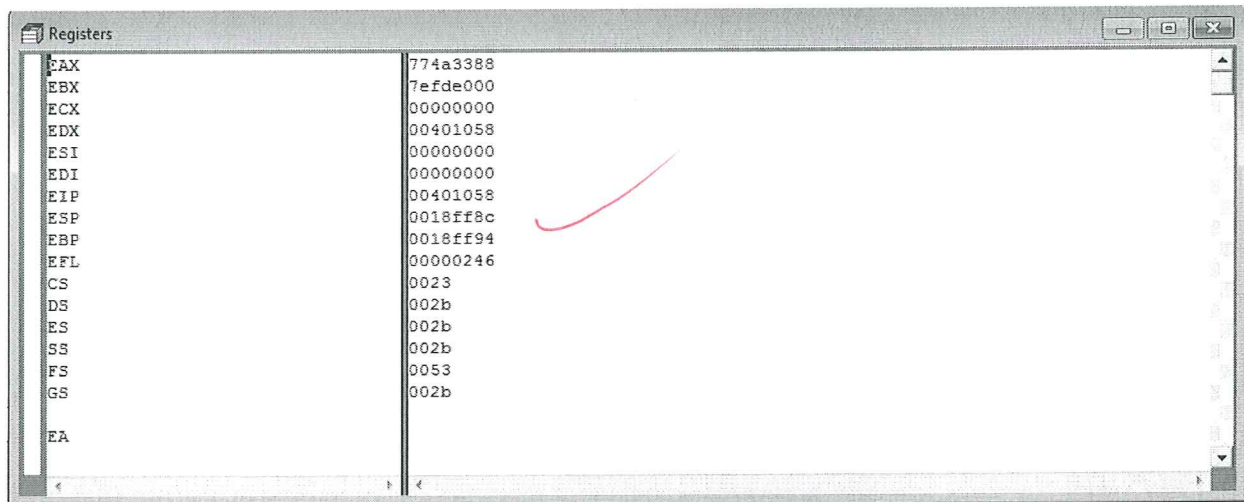
As soon as you type the T, go back to the code window. You are going to take a snapshot of the following:

- 1) the registers
- 2) Your source code with the yellow line on the cmp statement following the call getcharecho
- 3) the last 20h bytes of the stack
- 4) the contents of strInput

Get a printout of your screenshots, staple this sheet as a cover sheet, and submit it.

Task 1

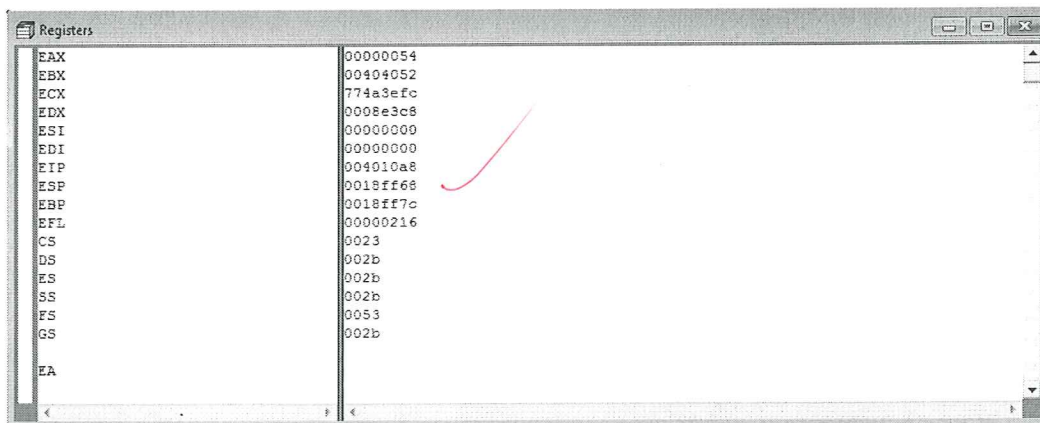
Registers



Register	Value
EAX	774a3388
EBX	7efde000
ECX	00000000
EDX	00401058
ESI	00000000
EDI	00000000
EIP	00401058
ESP	0018ff8c
EBP	0018ff94
EFL	00000246
CS	0023
DS	002b
ES	002b
SS	002b
FS	0053
GS	002b
EA	

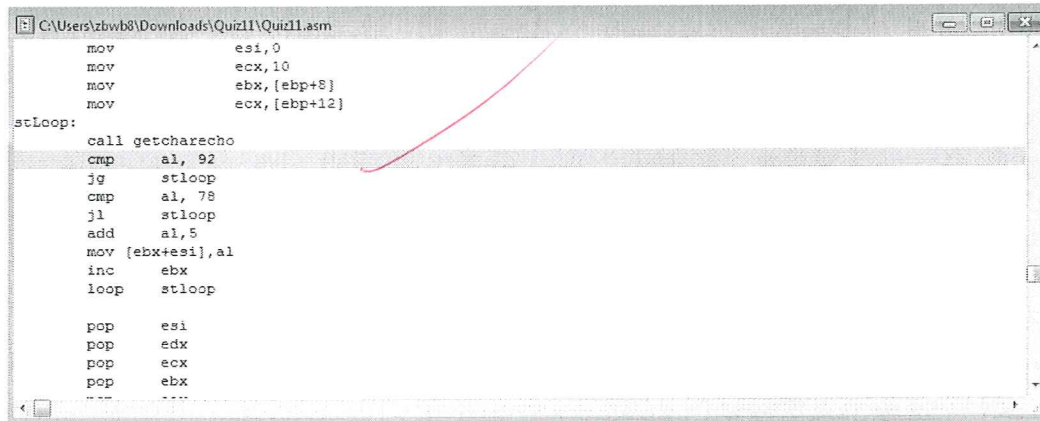
Task 2

Registers after string is entered



Register	Value
EAX	00000054
EBX	00404052
ECX	774a3efc
EDX	0008e3c8
ESI	00000000
EDI	00000000
EIP	004010a8
ESP	0018ff68
EBP	0018ff7c
EFL	00000216
CS	0023
DS	002b
ES	002b
SS	002b
FS	0053
GS	002b
EA	

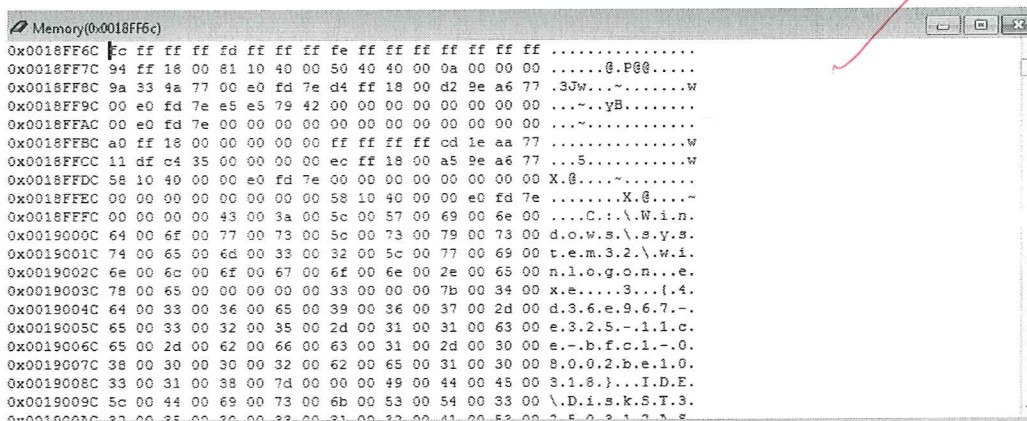
Yellow line on source code



```
C:\Users\zbwb8\Downloads\Quiz11\Quiz11.asm
mov     esi, 0
mov     ecx, 10
mov     ebx, [ebp+8]
mov     ecx, [ebp+12]
stLoop:
call    getcharecho
cmp     al, 92
jg      stloop
cmp     al, 78
jl      stloop
add     al, 5
mov     [ebx+esi], al
inc     ebx
loop    stloop

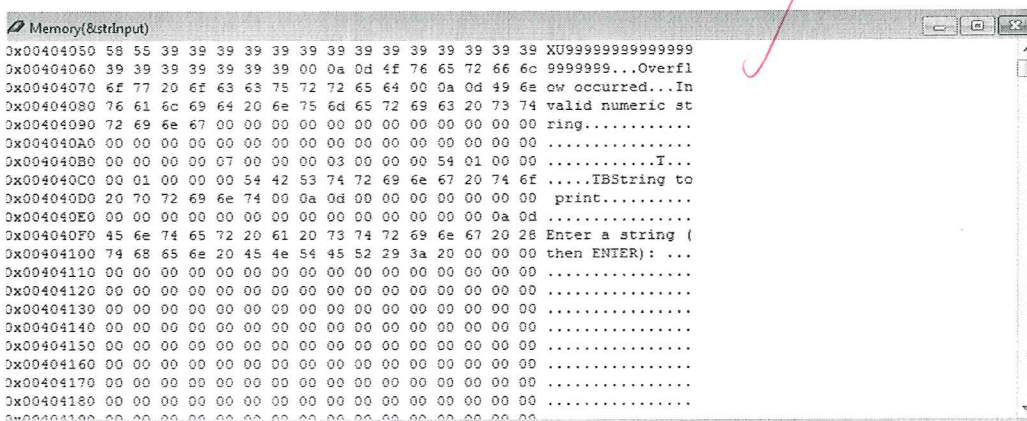
pop     esi
pop     edx
pop     ecx
pop     ebx
```

Last 20 bytes in the stack



```
Memory(0x0018FF6c)
0x0018FF6C ff ff ff fd ff ff ff ff ff ff ff ff ff ff ff ff .....
0x0018FF7C 94 ff 18 00 81 10 40 00 50 40 40 00 0a 00 00 00 .....@.P@....
0x0018FF8C 9a 33 4a 77 00 e0 fd 7e d4 ff 18 00 d2 9e a6 77 .3Jw....W
0x0018FF9C 00 e0 fd 7e e5 e5 79 42 00 00 00 00 00 00 00 ...~.yB.....
0x0018FFAC 00 e0 fd 7e 00 00 00 00 00 00 00 00 00 00 00 .....
0x0018FFBC a0 ff 18 00 00 00 00 00 ff ff ff ff cd 1e aa 77 .....W
0x0018FFCC 11 df c4 35 00 00 00 00 ec ff 18 00 a5 9e a6 77 ...5.....W
0x0018FFDC 58 10 40 00 00 e0 fd 7e 00 00 00 00 00 00 00 X. ....
0x0018FFEC 00 00 00 00 00 00 00 00 00 00 58 10 40 00 e0 fd 7e .....X.@....~
0x0018FFFC 00 00 00 00 43 00 3a 00 5c 00 57 00 69 00 6e 00 ...C...\.W.i.n.
0x0019000C 64 00 6f 00 77 00 73 00 5c 00 73 00 79 00 73 00 d.o.w.s.\.s.y.s.
0x0019001C 74 00 65 00 6d 00 33 00 32 00 5c 00 77 00 69 00 t.e.m.3.2.\.w.i.
0x0019002C 6e 00 6c 00 6f 00 67 00 6f 00 6e 00 2e 00 65 00 n.l.o.g.o.n...e.
0x0019003C 78 00 65 00 00 00 00 00 33 00 00 00 7b 00 34 00 x.e....3...{.4.
0x0019004C 64 00 33 00 36 00 65 00 39 00 36 00 37 00 2d 00 d.3.6.e.9.6.7.-.
0x0019005C 65 00 33 00 32 00 35 00 2d 00 31 00 31 00 63 00 e.3.2.5.-.1.1.c.
0x0019006C 65 00 2d 00 62 00 66 00 63 00 31 00 2d 00 30 00 e.-.b.f.c.1.-.0.
0x0019007C 38 00 30 00 30 00 32 00 62 00 65 00 31 00 30 00 8.0.0.2.b.e.1.0.
0x0019008C 33 00 31 00 38 00 7d 00 00 00 49 00 44 00 45 00 3.1.8}...I.D.E.
0x0019009C 5c 00 44 00 69 00 73 00 6b 00 53 00 54 00 33 00 \.D.i.s.k.S.T.3.
0x001900AC 23 00 35 00 30 00 32 00 31 00 32 00 41 00 52 00 3.5.0.2.1.2.3.2
```

Contents of strInput



```
Memory(&strInput)
0x00404050 58 55 39 39 39 39 39 39 39 39 39 39 39 39 39 39 XU9999999999999999
0x00404060 39 39 39 39 39 39 39 39 00 0a 0d 4f 76 65 72 66 6c 99999999...Overfl
0x00404070 6f 77 20 6f 63 63 75 72 72 65 64 00 0a 0d 49 6e ow occurred...In
0x00404080 76 61 6c 69 64 20 6e 75 6d 65 72 69 63 20 73 74 valid numeric st
0x00404090 72 69 6e 67 00 00 00 00 00 00 00 00 00 00 00 00 ring.....
0x004040A0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
0x004040B0 00 00 00 00 07 00 00 00 03 00 00 00 54 01 00 00 .....I...
0x004040C0 00 01 00 00 00 54 42 53 74 72 69 6e 67 20 74 6f ....IBString to
0x004040D0 20 70 72 69 6e 74 00 0a 0d 00 00 00 00 00 00 00 print.....
0x004040E0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 0a 0d .....
0x004040F0 45 6e 74 65 72 20 61 20 73 74 72 69 6e 67 20 28 Enter a string (
0x00404100 74 68 65 6e 20 45 4e 54 45 52 29 3a 20 00 00 00 then ENTER): ...
0x00404110 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
0x00404120 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
0x00404130 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
0x00404140 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
0x00404150 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
0x00404160 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
0x00404170 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
0x00404180 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
0x00404190 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
```