|  |  |  |
| --- | --- | --- |
| Week | Reverse Engineering Malware | Duration |
| 6 | Conditional Processing | 120 mins |

Marks allocation: 10/100 for CA tutorial submission

**Lesson Objectives**

* Understand Boolean and comparison instructions , conditional processing in assembly language

1. What will be the value of BX after the following instructions execute? Ans:6BH

mov bx,0FFFFh

and bx,6Bh

2. What will be the value of BX after the following instructions execute? Ans:92H

mov bx,91BAh

and bx,92h

3. What will be the value of BX after the following instructions execute? Ans:64bb

mov bx,0649Bh

or bx,3Ah

4. What will be the value of BX after the following instructions execute? Ans:A857H

mov bx,029D6h

xor bx,8181h

5. What will be the value of EBX after the following instructions execute?

Ans: BFAFF69Fh

mov ebx,0AFAF649Bh

or ebx,3A219604h

6. In the following instruction sequence, show the resulting value of AL where indicated, in binary: Ans: AL = 2Dh, 48h, 6Fh, A3h

mov al,01101111b

and al,00101101b ; a.

mov al,6Dh

and al,4Ah ; b.

mov al,00001111b

or al,61h ; c.

mov al,94h

xor al,37h ; d.

7. In the following instruction sequence, show the resulting value of AL where indicated, in hexadecimal: Ans: AL = 85h, 34h, BFh, AEh

mov al,7Ah

not al ; a.

mov al,3Dh

and al,74h ; b.

mov al,9Bh

or al,35h ; c.

mov al,72h

xor al,0DCh ; d.

8. In the following instruction sequence, show the values of the Carry, Zero, and Sign flags where indicated:

mov al,00001111b

test al,00000010b ; a. CF=0 ZF=0 SF=0

mov al,00000110b

cmp al,00000101b ; b. CF=0 ZF=0 SF=0

mov al,00000101b

cmp al,00000111b ; c. CF=1 ZF=0 SF=1

9. Implement the following pseudocode in assembly language. Assume that X is a 32-bit variable.

if( ebx > ecx ) OR ( ebx > val1 )

X = 1

else

X = 2

Ans:

main PROC

mov ebx,2

mov ecx,1

mov val1,2

cmp ebx,ecx

ja L1

cmp ebx,val1

ja L1

mov X,2

jmp exits

L1:

mov X,1

exits:

exit

main ENDP

10. Implement the following pseudocode in assembly language. Assume that A, B, and N are 32-bit signed integers.

while N > 0

if N != 3 AND (N < A OR N > B)

N = N – 2

else

N = N

Ans:

main PROC

mov eax, n

mov ebx, a

mov ecx, b

whileloop:

cmp eax, 0

jle exits

cmp eax, 3

je L3

jmp L1

L1:

cmp eax, ebx

jl L2

cmp eax, ecx

jg L2

jmp L3

L2:

sub eax, 2

jmp whileloop

L3:

sub eax, 1

jmp whileloop

exits:

exit

main ENDP

END