|  |  |  |
| --- | --- | --- |
| Week | Reverse Engineering Malware | Duration |
| 5 | Conditional Processing -1 | 120 mins |

Marks allocation: 10/100 for CA practical submission

**Lesson Objectives**

* Understand conditional processing in assembly language
* Write and execute assembly language programs using conditional processing

1. What will be the final value in EDX after this code executes?

mov edx,1

mov eax,7FFFh

cmp eax,8000h

jb L1

mov edx,0

L1:

**EDX = 1**

2. What will be the final value in EDX after this code executes?

mov edx,1

mov eax,7FFFh

cmp eax,0FFFF8000h

jl L2

mov edx,0

L2:

**EDX = 0**

3. (**True**/False): The following code will jump to the label named Target.

mov eax,-30

cmp eax,-50

jg Target

4. (**True/**False): The following code will jump to the label named Target.

mov eax,-42

cmp eax,26

ja Target

5.

Create a procedure named CalcGrade that receives an integer value between 0 and 100, and returns a single capital letter in the register. Preserve all other register values between calls to the procedure. The letter returned by the procedure should be according to the following ranges:

|  |  |
| --- | --- |
| Score Range | Letter Grade |
| 80 -100 | A |
| 70 - 79 | B |
| 60 - 69 | C |
| 50 - 59 | D |
| 0 - 49 | F |

**INCLUDE Irvine32.inc  
  
.data  
score DWORD 67  
grade byte ?  
  
.code  
main PROC  
    mov eax, score  
    cmp eax, 79  
    ja GradeA  
    cmp eax, 69  
    ja GradeB  
    cmp eax, 59  
    ja GradeC  
    cmp eax, 49  
    ja GradeD  
    cmp eax, 0  
    ja GradeF  
  
    GradeA:  
        mov grade, 'A'  
        exit  
    GradeB:  
        mov grade, 'B'  
        exit  
    GradeC:  
        mov grade, 'C'  
        exit  
    GradeD:  
        mov grade, 'D'  
        exit  
    GradeF:  
        mov grade, 'F'  
        exit  
  
  
     
main ENDP  
END main**

END