|  |  |  |
| --- | --- | --- |
| 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? Ans:1

mov edx,1

mov eax,7FFFh

cmp eax,8000h

jb L1

mov edx,0

L1

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

mov edx,1

mov eax,7FFFh

cmp eax,0FFFF8000h

jl L2

mov edx,0

L2:

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

mov eax,-30

cmp eax,-50

jg Target

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

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 |

Ans:

INCLUDE Irvine32.inc

.data

score SDWORD ?,0

titleScore BYTE "Score:",0

titleGrade BYTE " Grade:",0

grade BYTE ?,0

grA BYTE "A",0

grB BYTE "B",0

grC BYTE "C",0

grD BYTE "D",0

grF BYTE "F",0

out\_of\_range BYTE "The integer is not <= 100 and >= 0",0

.code

main PROC

call Randomize

mov eax,101 ;set random range from 0 to 100

mov edx,OFFSET titleScore

call WriteString

;call RandomRange

mov eax, 1000

mov score, eax

mov edx, score

call writeDec

mov edx,OFFSET titleGrade

call WriteString

call GradeCalc

mov edx,0

mov grade,al

mov edx, OFFSET grade

call WriteString

exit

main ENDP

GradeCalc PROC

cmp eax,101

jl Check1

jnl Error1

Check1:

cmp eax,80

jl Check2

mov al,grA

ret

Check2:

cmp eax,70

jl Check3

mov al,grB

ret

Check3:

cmp eax,60

jl Check4

mov al,grC

ret

Check4:

cmp eax,50

jl Check5

mov al,grD

ret

Check5:

cmp eax,0

jl Error1

mov al,grF

ret

Error1:

mov edx,OFFSET out\_of\_range

call WriteString

call Crlf

ret

GradeCalc ENDP

END main

END