**Compare unsigned EAX to EBX, and copy the larger of the two into a variable named Large**:

%include "io.inc"

section .data

Large dd 0 ; Variable to store the larger value (32-bit double word)

section .text

global CMAIN ; Entry point for the program

CMAIN:

mov ebp, esp ; Set up the base pointer for debugging

; Load example values into registers

mov eax, 10 ; EAX = 10

mov ebx, 20 ; EBX = 20

; Compare EAX and EBX

cmp eax, ebx ; Compare EAX and EBX

jge larger\_eax ; If EAX >= EBX, jump to larger\_eax

mov eax, ebx ; If EBX is larger, move EBX to EAX

larger\_eax:

mov [Large], eax ; Store the larger value (EAX or EBX) in 'Large'

; Print the larger value

PRINT\_DEC 4, [Large] ; Print the value stored in 'Large'

NEWLINE

; Exit the program

xor eax, eax ; Clear EAX for return value (0)

ret ; Return from CMAIN

**Jump to label L1 if the memory word pointed to by ESI equals zero:**

include "io.inc"

section .data

value dd 0 ; Define a 32-bit value initialized to 0

section .text

global CMAIN

CMAIN:

mov ebp, esp ; for correct debugging

; Load the address of value into ESI

mov esi, value

mov eax, [esi] ; Load the value at [ESI] into EAX

; Compare the value with 0

cmp eax, 0

je L1 ; Jump if equal to L1

; If not zero, skip to end

jmp end\_program

L1:

; Code for label L1

mov eax, 1 ; Set EAX to 1

end\_program:

; Store final result in eax

PRINT\_DEC 4, eax ; Print the value in EAX

NEWLINE

xor eax, eax ; Return 0

ret ; Return from CMAIN