





section .data

msg\_prompt db 'Enter a digit (0-9): ', 0

msg\_below db 'Input is below 5', 10, 0

msg\_equal db 'Input is equal to 5', 10, 0

msg\_above db 'Input is above 5', 10, 0

section .bss

input resb 2 ; Reserve 2 bytes for input

section .text

global \_start

\_start:

; Print prompt

mov eax, 4 ; sys\_write

mov ebx, 1 ; stdout

mov ecx, msg\_prompt

mov edx, 21

int 0x80

; Read 1 byte from user

mov eax, 3 ; sys\_read

mov ebx, 0 ; stdin

mov ecx, input

mov edx, 1

int 0x80

; Convert ASCII to number

movzx eax, byte [input]

sub eax, '0'

; Compare to 5

cmp eax, 5

je equal

jl below

jg above

below:

call print\_below

jmp exit

equal:

call print\_equal

jmp exit

above:

call print\_above

exit:

mov eax, 1

xor ebx, ebx

int 0x80

print\_below:

mov eax, 4

mov ebx, 1

mov ecx, msg\_below

mov edx, 19

int 0x80

ret

print\_equal:

mov eax, 4

mov ebx, 1

mov ecx, msg\_equal

mov edx, 21

int 0x80

ret

print\_above:

mov eax, 4

mov ebx, 1

mov ecx, msg\_above

mov edx, 20

int 0x80

ret

**EXPLANATION OF THE PROGRAM**

The three primary sections of the program are text, bss, and data. Predefined strings used for output and prompting messages are stored in the.data section. The user's input is reserved in the.bss section. The application begins by reading a single-digit character from the user and displaying a prompt in the.text section. After that, it uses the cmp instruction to convert the ASCII input into a numeric value and compare it to 5. It jumps to the relevant label to invoke a particular procedure (print\_below, print\_equal, or print\_above) based on the comparison. Every procedure prints its message using a system call, and the program terminates with an exit syscall.