Q#1 Practice Basic Codes for `MUL` and `DIV`

<u>Multiplication (`MUL`):</u>

```
section .data

   num1 db 5

   num2 db 6

   result dw 0


section .text

   global _start


_start:

   mov al, [num1]       ; Load num1 into AL

   mov bl, [num2]       ; Load num2 into BL

   mul bl               ; Multiply AL by BL (AL = AL * BL)

   mov [result], ax     ; Store result in memory


   ; Exit

   mov ah, 0x4C

   int 0x21
```

<u>Division (`DIV`):</u>

```
section .data

   dividend db 12

   divisor db 4

   quotient db 0

   remainder db 0


section .text

   global _start
```

```
_start:
    mov al, [dividend]   ; Load dividend into AL
    mov bl, [divisor]    ; Load divisor into BL
    div bl               ; Divide AL by BL (AL = AL / BL)

    mov [quotient], al   ; Store quotient
    mov [remainder], ah  ; Store remainder

    ; Exit
    mov ah, 0x4C
    int 0x21
```

Q#2 Program to Check if a Number is Even or Odd

```
section .data
    prompt_msg db 'Enter a number: $'
    even_msg db 'The number is EVEN.$'
    odd_msg db 'The number is ODD.$'
    buffer db 50, 0

section .text
    global _start

_start:
    mov ah, 0x09
    lea dx, [prompt_msg]
    int 0x21

    mov ah, 0x0A
    lea dx, [buffer]
    int 0x21

    mov al, [buffer + 1]   ; Get the input number
    sub al, 48             ; Convert ASCII to number
    and al, 1              ; Check the least significant bit (even = 0, odd = 1)
    jz even

odd:
    mov ah, 0x09
    lea dx, [odd_msg]
    int 0x21
    jmp exit

even:
    mov ah, 0x09
    lea dx, [even_msg]
    int 0x21

exit:
    mov ah, 0x4C
    int 0x21
```