ejercicio 2

```
section .data
  msg_positivo db 'El numero es positivo', 0xA, 0
  msg_negativo db 'El numero es NEGATIVO', 0xA, 0
  msg_cero db 'El numero es cero', 0xA, 0

section. bss

num resb 1 ; reservar un byte para el numero

section .text
  global _start

_start:

  ;LEER EL numero
  ;CODIGO DE LECTURA

  ;CLASICAR EL numero
  mov al, [num] ; cargar el numero en AL
  cmp al, 0

  je es_cero   ; si es igual a cero saltar a es_cero
  jl es_negativo ; si es menor que cero saltar a es_negativo
  jg es_positivo ; si es mayor que cero, saltar a es_positivo

  es_cero:
  mov ecx, msg_cero
  jmp print_result

  es_negativo:
  mov ecx, msg_negativo
  jmp print_result

  es_positivo:
  mov ecx, msg_positivo
  jmp print_result

  print_result
  ; codigo para imprimir el mensaje

```

ejercicio 3. Par o Impar: Leer un número y determinar si es par o impar usando únicamente la bandera de paridad (PF).

```
section .data
    msg_par   db 'El numero es PAR', 0xA, 0
    msg_impar db 'El numero es IMPAR', 0xA, 0

section .bss
    num resb 1

section .text
    global _start

_start:
    ; Leer el número
    ; (Código de lectura aquí...)

    mov al, [num]     ; Cargar el número en AL
    test al, 1        ; Verifica el bit menos significativo

    jp es_par         ; Si PF=1 (resultado tiene bits en 1 en cantidad par), es PAR
    jmp es_impar      ; Si no, es IMPAR

es_par:
    mov ecx, msg_par
    jmp print_result

es_impar:
    mov ecx, msg_impar
    jmp print_result

print_result:
    ; (Código para imprimir el mensaje)
    ; (Código de salida aquí...)
```