

Ejercicio 2 (3 puntos)

Dado el siguiente programa en ASM que quiere ser compilado como programa principal y linkeditado con las dependencias necesarias, encuentre al menos 3 errores:

```
section .data
number db "%d", 10, 0 ; declaro el formato para imprimir

section .text
global main
extern printf ; Usamos la función standard printf

start:
mov ecx, 9999 ; Mueve el número hasta el cual vamos a buscar
palindromos ; este programa no puede usar números mayores a 9999
empieza:
push ecx ; Guarda el tope en el stack
mov ax, cx ; Lleva el número a AX
mov bx, 10 ; Guarda 10 en BX para dividir por 10
mov cx, 0 ; Reseteo CX para la próxima operación

pdigits: ; dividimos por 10 para averiguar la cantidad
; de dígitos del número
mov dx, 0 ; llevamos DX a 0, usaremos DX para guardar el módulo
; luego de la división
div bx ; hacemos AX/BX, AX contiene el número que
; estamos probando si es palindromo
push dx ; Push DX, los dígitos a la derecha al stack
inc cx ; Incrementamos CX, CX llevará la cuenta de la
; cantidad de dígitos
cmp ax, 0 ; si es 0 ya llegamos
je cont ; si es así vamos al rótulo 'cont'
jmp pdigits ; sino el dígito de la derecha e incrementamos el contador

cont:
cmp cx, 4 ; si el número contiene 4 dígitos
je cuatro ; vamos a 'cuatro'
cmp cx, 3 ; si el número contiene 3 dígitos
je tres ; vamos a 'tres'
cmp cx, 2 ; si el número contiene 2 dígitos
je two ; vamos a 'dos'
cmp cx, 1 ; si el número contiene 1 dígito
je one ; vamos a 'uno'

cuatro:
; si el número contiene 4 dígitos
```

declarado mal → debería decir main

0010 0111 0000 1111 ← ocupa 16 bits válido para cx

falta ;

9
9
9
9
9999
ret so

llama un rótulo que no existe

ret start
%
9999
ret so

se va * luego de ejecutar

número ya estaba en stack en orden para printf

mal desreferencia puntero cuando necesitas un puntero no el primer char

mal porque al hacer add sp, 8 ante su pila "libero valores" de parametro printf y ya no hay mas nada que el ret so

```
pop ax = 9 ; el número tiene 4 dígitos
pop bx = 9 ; recuperamos los 4 dígitos en AX, BX, CX y DX.
pop cx = 9 ; o sea que si el número es por ej 1221 los registros serian
pop dx = 9 ; AX->1, BX->2, CX->2 y DX->1.
cmp ax, dx ; en ese caso si (AX == DX) y (BX == CX) entonces
; el número es palindromo
jne nope ; vamos a 'yep' sino a 'nope'
cmp bx, cx ;
jne nope ;
je yep ; en forma similar en 'tres' y 'dos'

tres:
pop ax
pop bx
pop cx
cmp ax, cx
jne nope
je yep

dos:
pop ax
pop bx
cmp ax, bx
jne nope
je yep

uno:
pop ax ; si el número es de un solo dígito se realiza un solo pop
jmp yep ; no hay necesidad de analizar ese valor ya que todos
; los números de un solo dígito son palindromos
; se salta a 'yep'

nope:
pop ecx ; recuperamos ecx que es el valor actual
dec ecx ; lo decrementamos para continuar con el número
siguiente ; falta
jnz empieza ; comenzamos de vuelta

yep:
pop ecx ; si es palindromo recuperamos en ecx el valor original.
push ecx ; push para usarlo como argumento a printf

push [number] ; push del formato para printf
call printf ; Imprimimos
add sp, 8 ; Restauramos el stack pointer
; es sintaxis de asm intel sp debería ser esp
pop ecx ; print cambio ecx, lo recuperamos
push ecx ; push del número palindromo.
dec ecx ; vamos al siguiente número
jnz empieza ; seguimos
```

mal pop libera char de [number] o sea (%)