

Prácticas en NASM para Secundaria

```
; -----  
; 1. Suma de dos números (12 + 34)  
; -----  
section .data  
    msg db "Resultado: ", 0  
    salto db 10, 0  
  
section .bss  
    buffer resb 2    ; para almacenar decenas y unidades  
  
section .text  
    global _start  
  
_start:  
    mov eax, 12  
    add eax, 34      ; eax = 46  
  
    mov ebx, 10  
    xor edx, edx  
    div ebx          ; eax = 4 (decenas), edx = 6 (unidades)  
  
    add eax, '0'  
    mov [buffer], al ; decenas como ASCII  
    add edx, '0'  
    mov [buffer+1], dl ; unidades como ASCII  
  
    ; Mostrar mensaje  
    mov eax, 4  
    mov ebx, 1  
    mov ecx, msg  
    mov edx, 10  
    int 0x80  
  
    ; Mostrar número (2 dígitos)  
    mov eax, 4  
    mov ebx, 1  
    mov ecx, buffer  
    mov edx, 2  
    int 0x80  
  
    ; Salto de línea  
    mov eax, 4  
    mov ebx, 1  
    mov ecx, salto  
    mov edx, 1  
    int 0x80  
  
    ; Salir  
    mov eax, 1  
    xor ebx, ebx
```

Prácticas en NASM para Secundaria

```
int 0x80

; Compilar y ejecutar:
; nasm -f elf32 suma.asm -o suma.o
; ld -m elf_i386 suma.o -o suma
; ./suma

; -----
; 2. Conversión de binario a decimal (0b1010 = 10)
; -----

section .data
    bin_msg db "Binario 1010 = ", 0

section .bss
    bin_buffer resb 2

section .text
    global _start_bin

_start_bin:
    mov eax, 0b1010      ; 10 decimal
    mov ebx, 10
    xor edx, edx
    div ebx              ; eax = 1, edx = 0
    add eax, '0'
    mov [bin_buffer], al ; decena
    add edx, '0'
    mov [bin_buffer+1], dl ; unidad

    ; Mostrar texto
    mov eax, 4
    mov ebx, 1
    mov ecx, bin_msg
    mov edx, 17
    int 0x80

    ; Mostrar número
    mov eax, 4
    mov ebx, 1
    mov ecx, bin_buffer
    mov edx, 2
    int 0x80

    ; Salto de línea
    mov eax, 4
    mov ebx, 1
    mov ecx, salto
    mov edx, 1
    int 0x80

    ; Salir
```

Prácticas en NASM para Secundaria

```
mov eax, 1  
xor ebx, ebx  
int 0x80
```