

TALLER #9

<https://onecompiler.com/assembly/445rnv3r5>

```
section .data
    num1 db 5
    num2 db 12
    result db 0
    msg db 'Resultado: ', 0

section .bss
    buffer resb 4

section .text
    global _start

_start:

    mov al, [num1]
    add al, [num2]
    mov [result], al
    ; Convertir el resultado a ASCII
    movzx eax, byte [result]
    add eax, 48 ; Convertir el valor numérico en su correspo
    mov [buffer], al ; Almacenar el carácter ASCII en el buf
```

Output:

Resultado: A

g. Utilizando de nuevo el código de prueba original, modifica el código para que ahora utilice el modo de direccionamiento inmediato e indirecto (en programas separados) para que imprima el carácter '@'. Documenta tus resultados.

INMEDIATO

```
section .data
    result db 0
    msg db 'Resultado: ', 0

section .bss
    buffer resb 4

section .text
    global _start

_start:

    mov al, 5
    add al, 11
    mov [result], al
; Convertir el resultado a ASCII
    movzx eax, byte [result]
    add eax, 48 ; Convertir el valo
    mov [buffer], al ; Almacenar el
```

INDIRECTO

```
section .data
    num1 db 5
    num2 db 11
    result db 0
    msg db 'Resultado: ', 0
```

```
section .bss
    buffer resb 4
```

```
section .text
    global _start
```

_start:

```
    mov esi, num1      ; ESI -> num1
    mov al, [esi]       ; Obtener el valor de num1
```

```
    mov edi, num2      ; EDI -> num2
    add al, [edi]       ; Sumar el valor de num2 a num1
```

```
    mov [result], al
```