

```

1 section .data
2     num1 db 5
3     num2 db 11
4     result db 0           ; Reserva un byte para almacenar el resultado de la suma
5     message db "Resultado: ", 0 ; Cadena de texto que se imprimirá antes del resultado
6
7 section .bss
8     buffer resb 4         ; Reserva espacio de 4 bytes en buffer (para almacenar el número como texto)
9
10 section .text
11     global _start
12
13 %macro PRINT_STRING 1
14     ; Macro para imprimir una cadena de texto
15     mov eax, 4             ; Número de la llamada al sistema (sys_write)
16     mov ebx, 1             ; File descriptor 1 (stdout)
17     mov ecx, %1            ; Dirección de la cadena a imprimir (pasada como parámetro a la macro)
18     mov edx, 13            ; Longitud de la cadena (en este caso, "Resultado: " tiene 13 caracteres)
19     int 0x80              ; Interrupción para realizar la llamada al sistema
20 %endmacro
21
22 %macro PRINT_NUMBER 1
23     ; Macro para imprimir un número (suponiendo que es de un solo dígito)
24     mov eax, %1            ; Cargar el número en eax
25     add eax, '0'           ; Convertir el número a su valor ASCII sumando el valor de '0' (48)
26     mov [buffer], eax      ; Almacenar el valor ASCII en buffer
27     mov eax, 4             ; Número de la llamada al sistema (sys_write)
28     mov ebx, 1             ; File descriptor 1 (stdout)
29     mov ecx, buffer        ; Dirección del buffer (donde está el número en formato ASCII)
30     mov edx, 1             ; Solo un byte a imprimir (un solo dígito)
31     int 0x80              ; Interrupción para realizar la llamada al sistema
32 %endmacro
33
34 _start:
35     ; Cálculo de la suma
36     mov al, [num1]         ; Cargar el valor de num1 en el registro AL (registro de 8 bits)
37     add al, [num2]         ; Sumar el valor de num2 a AL (AL = AL + num2)
38     mov [result], al       ; Almacenar el resultado de la suma en la variable result
39
40     ; Imprimir el mensaje "Resultado: "
41     PRINT_STRING message ; Llamar a la macro para imprimir la cadena "Resultado: "
42
43     ; Imprimir el número (el resultado de la suma)
44     PRINT_NUMBER [result] ; Llamar a la macro para imprimir el número almacenado en result
45
46     ; Salir del programa
47     mov eax, 1             ; Número de la llamada al sistema (sys_exit)
48     mov ebx, 0             ; Código de salida 0 (sin errores)
49     int 0x80              ; Interrupción para realizar la llamada al sistema
50

```

```

1 section .data
2     message db "La suma de los valores es: ", 0
3     newline db 10, 0           ; Nueva línea para la salida
4
5 section .bss
6     buffer resb 4               ; Buffer para convertir números a caracteres
7
8 section .text
9     global _start
10
11 %macro DEFINE_VALUES 3
12     ; Define una "estructura" con tres valores
13     val1 db %1                 ; Primer valor
14     val2 db %2                 ; Segundo valor
15     val3 db %3                 ; Tercer valor
16 %endmacro
17
18 %macro PRINT_STRING 1
19     mov eax, 4                 ; Syscall número para 'write'
20     mov ebx, 1                 ; File descriptor para stdout
21     mov ecx, %1                ; Dirección del mensaje
22     mov edx, 25                ; Longitud del mensaje
23     int 0x80
24 %endmacro
25
26 %macro PRINT_NUMBER 1
27     ; Convierte un número en eax a caracteres ASCII y lo imprime
28     mov eax, %1
29     mov ecx, buffer            ; Usamos el buffer para guardar el resultado
30     mov ebx, 10                ; Divisor para obtener dígitos decimales
31
32 .next_digit:
33     xor edx, edx               ; Limpia edx para la división
34     div ebx                    ; Divide eax entre 10, cociente en eax, residuo en edx
35     add dl, '0'                ; Convierte el dígito a ASCII
36     dec ecx                    ; Mueve hacia atrás en el buffer
37     mov [ecx], dl              ; Almacena el dígito en el buffer
38     test eax, eax              ; Verifica si quedan dígitos
39     jnz .next_digit            ; Si quedan dígitos, continúa
40
41     ; Imprime el número
42     mov eax, 4                 ; Syscall para write
43     mov ebx, 1                 ; Salida estándar
44     mov ecx, buffer            ; Comienza en el primer dígito
45     mov edx, buffer + 4        ; Longitud máxima asumida en 4 dígitos
46     sub ecx, edx               ; Calcula la longitud real
47     int 0x80
48 %endmacro
49
50 %macro PRINT_SUM 0
51     ; Realiza la suma de tres valores y la imprime
52     mov al, [val1]             ; Carga el primer valor en AL
53     add al, [val2]             ; Suma el segundo valor
54     add al, [val3]             ; Suma el tercer valor
55     movzx eax, al              ; Expande AL a EAX para asegurar un valor de 32 bits
56
57     ; Imprime el resultado de la suma
58     PRINT_NUMBER eax
59     PRINT_STRING newline
60 %endmacro
61
62 ; Definimos los tres valores con la macro DEFINE_VALUES
63 DEFINE_VALUES 3, 5, 7
64
65 _start:
66     ; Imprime el mensaje inicial
67     PRINT_STRING message
68
69     ; Imprime la suma de los valores
70     PRINT_SUM
71
72     ; Salir del programa
73     mov eax, 1                 ; Syscall para 'exit'
74     mov ebx, 0                 ; Código de salida
75     int 0x80
76

```

Input for the program (Optional):

Output:

La suma de los valores es
15