



**Universidad Autónoma de Baja California
Facultad de Ingeniería Arquitectura y Diseño**



Ingeniería en Software y Tecnologías Emergentes

Materia: Organización de Computadoras

Alumno: Jesus Eduardo Rodríguez Ramírez

Profesor: Jonatan Crespo Ragland

Grupo 932

Trabajo: Taller 12

Ensenada, B.C; a 26 de noviembre del 2024

Desarrollar un programa en ensamblador de su elección. Tratando de utilizar todo lo visto en clase.

```
1 section .data
2     mensajeBase db "Ingrese la base del triangulo: ", 0
3     mensajeAltura db "Ingrese la altura del triangulo: ", 0
4     mensajeResultado db "El area del triangulo es: ", 0
5     buffer db 20, 0 ; Buffer para el resultado convertido a texto
6
7 section .bss
8     base resb 4 ; Espacio para almacenar La base
9     altura resb 4 ; Espacio para almacenar La altura
10    area resb 4 ; Espacio para almacenar el área calculada
11
12 section .text
13     global _start
14
15 _start:
16     ; Solicitar La base
17     mov eax, 4 ; syscall write
18     mov ebx, 1 ; salida estándar
19     mov ecx, mensajeBase ; mensaje a imprimir
20     mov edx, 30 ; longitud del mensaje
21     int 0x80
22
23     ; Leer La base
24     mov eax, 3 ; syscall read
25     mov ebx, 0 ; entrada estándar
26     mov ecx, base ; buffer para La entrada
27     mov edx, 4 ; longitud máxima
28     int 0x80
29
30     ; Convertir La base a número
31     mov ecx, base ; Dirección de La entrada
32     call atoi ; Convierte texto a número (resultado en EAX)
33     mov [base], eax ; Guardar el número convertido
34
35     ; Solicitar La altura
36     mov eax, 4 ; syscall write
37     mov ebx, 1 ; salida estándar
38     mov ecx, mensajeAltura ; mensaje a imprimir
39     mov edx, 31 ; longitud del mensaje
40     int 0x80
41
42     ; Leer La altura
43     mov eax, 3 ; syscall read
44     mov ebx, 0 ; entrada estándar
45     mov ecx, altura ; buffer para La entrada
46     mov edx, 4 ; longitud máxima
47     int 0x80
48
49     ; Convertir La altura a número
50     mov ecx, altura ; Dirección de La entrada
51     call atoi ; Convierte texto a número (resultado en EAX)
52     mov [altura], eax ; Guardar el número convertido
53
54     ; Calcular el área (base * altura / 2)
55     mov eax, [base] ; Cargar La base
56     imul eax, [altura] ; Multiplicar base * altura
57     shr eax, 1 ; Dividir entre 2 (desplazamiento lógico)
58     mov [area], eax ; Guardar el área calculada
59
60     ; Convertir el área a texto
61     mov eax, [area] ; Cargar el valor del área
```

```

60 ; Convertir el área a texto
61 mov eax, [area] ; Cargar el valor del área
62 mov ecx, buffer ; Buffer para almacenar el texto
63 call itoa ; Convertir número a texto
64
65 ; Imprimir el mensaje del resultado
66 mov eax, 4 ; syscall write
67 mov ebx, 1 ; salida estándar
68 mov ecx, mensajeResultado ; mensaje a imprimir
69 mov edx, 27 ; Longitud del mensaje
70 int 0x80
71
72 ; Imprimir el área calculada
73 mov eax, 4 ; syscall write
74 mov ebx, 1 ; salida estándar
75 mov ecx, buffer ; Buffer con el resultado
76 mov edx, 20 ; Longitud máxima del buffer
77 int 0x80
78
79 ; Terminar el programa
80 mov eax, 1 ; syscall exit
81 xor ebx, ebx ; Código de salida 0
82 int 0x80
83
84 ; Subrutina: atoi (convierte texto ASCII a entero)
85 ▾ atoi:
86 xor eax, eax ; Limpiar acumulador
87 xor ebx, ebx ; Limpiar EBX
88 ▾ .parse_loop:
89 mov bl, byte [ecx] ; Leer carácter actual
90 cmp bl, 10 ; ¿Salto de línea?
91 je .done ; Si es salto de línea, terminar
92 sub bl, '0' ; Convertir ASCII a valor numérico
93 imul eax, eax, 10 ; Multiplicar acumulador por 10
94 add eax, ebx ; Sumar el dígito
95 inc ecx ; Avanzar al siguiente carácter
96 jmp .parse_loop ; Repetir
97 ▾ .done:
98 ret
99
100 ; Subrutina: itoa (convierte entero a texto ASCII)
101 ▾ itoa:
102 xor ebx, ebx ; Limpiar EBX
103 mov edx, 0 ; Limpiar EDX
104 mov ebx, 10 ; Base decimal
105 ▾ .reverse_digits:
106 xor edx, edx ; Limpiar resto
107 div ebx ; Dividir EAX entre 10
108 add dl, '0' ; Convertir dígito a ASCII
109 dec ecx ; Retroceder en el buffer
110 mov [ecx], dl ; Guardar el dígito en el buffer
111 test eax, eax ; ¿EAX es 0?
112 jnz .reverse_digits ; Si no, repetir
113 ret
114

```

```

Ingrese la base del triangulo:10
Ingrese la altura del triangulo:20
El area del triangulo es:100 ..

```