Consulta MA-1006: 3/04/2025.

Ejercicios que vamos a trabajar (de la lista de ejercicios):

MATLAB: #9, #13, #17.

Método de Bisección: #3, #5.
Método del Punto Fijo: #2, #4.

MATLAB:
Ejercicio (Adicional). Cree una funión "promV" que reciba como entrada un vector v de tamaño n y devuelva el promedio de todas las entradas de v en valor absoluto.

```
function prom = promV(v)
% Entradas:
% Salida:
n = length(v); % cantidad de elementos de v
W = abs(v);
prom= 0; % valor inicial
for i=1:n
    prom = prom + W(i);
end
prom = prom/n;
```

$v = (v_1, v_2, \dots, v_n)$

$W = (|v_1|, \dots, |v_n|)$

`[n,m]= size(A)`

Ejercicio #9. La siguiente M-función determina cuando un número es un cuadrado perfecto cuando un número es un cuadrado perfecto.

```
function [ ] = CuadradoPerfecto(n)
% Nos dice si n es cuadrado perfecto o no.
m = sqrt(n);   % raíz cuadrada de n
if floor(m) == m     % floor función parte entera
                      para comparar
      disp("es cuadrado perfecto");
else
      disp("No es cuadrado perfecto);
 end
end
```

Ejercicio #13. Escriba una M-función "AreaTri" que calcule el área de un triángulo a partir de las longitudes de sus lados: a, b, c, usando:

$$A = \sqrt{p(p-a)(p-b)(p-c)}\,, \quad \text{con } p = \frac{a+b+c}{2}$$

La función debe emitir mensaje de error en los siguientes casos:

a) Si alguna de las longitudes: a, b, c es menor o igual a cero.

b) Si el radicando es negativo.

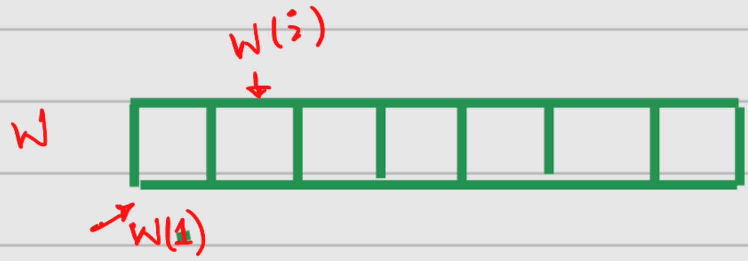
```
function A = AreaTri(a,b,c)
% Entradas:
% Salidas:
 if (a<=0) || (b<=0) || (c<=0)
     disp("Error! Se debe ingresar valores
                 mayores que cero");
     return ; % detenemos la ejecución.
 end
 p = (a+b+c)/2;
 rad = p*(p-a)*(p-b)*(p-c);

 if rad <= 0
     disp ("El radicando es menor que cero")
     return;
 end

 A = sqrt(rad);
end
```

```
function A = AreaTri (a,b,c)
% Entradas:
% Salidas :
p = (a+b+c)/2;
rad = p*(p-a)*(p-b)*(p-c);

if (a<=0) || (b<=0) || (c<=0)
    disp("Error!")
```

```
elseif (rad <= 0)
    disp("Error de nuevo")
else
    A = sqrt(rad);
end
```

Ejercicio #17. Escriba una M-función function [vmax] = Mayor(v) que reciba como argumento un vector v y devuelva el máximo entre los valores absolutos de todos sus componentes.

```
function [vmax] = Mayor(v)
% Entradas:
% Salida:
n = length(v); % Cantidad de elementos de v
w = abs(v);
vmax = w(1);

for i = 2:n
    if vmax < w(i)
        vmax = w(i);
    end
end
```



Para verificar:

```
v = rand(10,1)
Mayor(v)
```

Ejercicio Bisección #2. Encuentre la tercera iteración de la función:

$$f(x) = \sin(x) - \cos(1+x^2) - 1$$

en el intervalo $[1.8, 2]$.

Solución: f es continua.

`f=@(z) sin(z)-cos(1+z^2)-1;`

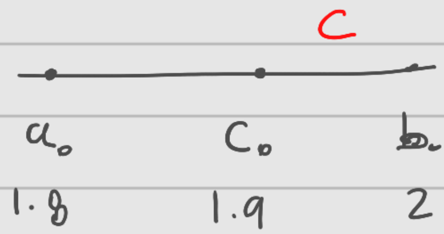
$f(1.8) \approx 0.4289$

$f(2) \approx -0.3743$

$\Big\}$ Por T.V.I existe $c \in [1.8, 2]$ tal que $f(c) = 0$.

1) $[a_0, b_0] = [1.8, 2]$

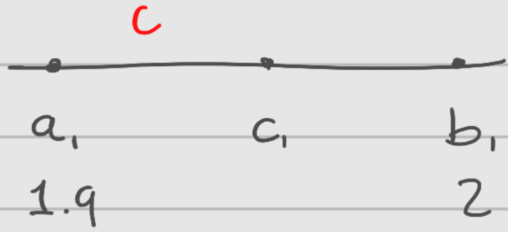
$$c_0 = \frac{(1.8+2)}{2} = 1.9$$



$f(c_0) \approx 0.0485$

2) $[a_1, b_1] = [1.9, 2]$

$$c_1 = \frac{(1.9+2)}{2} = 1.95$$



$f(c_1) \approx -0.1610$

3) $[a_2, b_2] = [1.9, 1.95]$

$$c_2 = \frac{(1.9+1.95)}{2} = 1.925$$

$f(x) = x - 2^{-x}$

Ejercicio #3 (Bisección). Considere la ecuación $x = 2^{-x}$. Proporcione un intervalo que contenga la solución de dicha ecuación. Además determine el número de iteraciones necesarios para alcanzar una tolerancia de 10^{-5}.

Solución: Defina $f(x) = x - 2^{-x}$

```
f = @(z) z-2.^(-z);
x = linspace(-2, 2, 100);
y = f(x);
```

plot (x,y)

Considere el intervalo $[0.5, 1.5]$
$[a, b]$

$f(1/2) = -0.2071$
$f(3/2) = 1.1464$ } Por T.V.I existe $c \in [0.5, 1.5]$ tal que $f(c) = 0$.

Por teorema:

$$\underbrace{|c - c_k|}_{} \leq \boxed{\underbrace{\frac{b-a}{2^k}}_{\leq 10^{-5}}}$$

Hacemos:

$$\frac{b-a}{2^k} \leq 10^{-5}$$

$$2^{-k} \leq 10^{-5}$$

$$\log_{10} 2^{-k} \leq \cancel{\log_{10}} 10^{-5}$$

$$-k \cdot \log_{10} 2 \leq -5$$

$$k \geq \frac{5}{\log_{10} 2} \approx 16.6096$$

Basta realizar 17 iteraciones.