TP Series No. 1
(SOLVING NON-LINEAR EQUATIONS)

Exercise 1 :

Consider the function $f(x) = x^4 + x - 1$.

1. Draw the graph of the function f on the interval $[-2, 2]$.
2. Apply the Matlab function 'bissection.m' on f, with: $eps = 0.01$.

Program: bissection.m

Inputs	Outputs
$\boldsymbol{f}$: Function to consider	$\boldsymbol{x}$: Root obtained by the method
a and **b** : Limits of the interval $[\mathbf{a}, \mathbf{b}]$	**y** : Image of the root by the function $\boldsymbol{f}$
eps: Given precision (Stopping criterion)	niter : Number of iterations performed

```
function[x,y,niter]=bissection(f,a,b,eps)
if f(a)*f(b)>0
error('Inappropriate interval');
return
end
niter=0;
while abs(b-a)>eps
x = (a+b)/2;
if f(x)==0
niter = niter +1; y=0;
return
elseif f(a)*f(x) > 0
a = x;
elseif f(b)*f(x) > 0
b = x;
end
niter = niter +1;
end
y=f(x);
end
```

Exercise 2 :

Let the function $f(x) = \frac{e^{-x} - x^2}{4}$.

- Apply the Matlab function 'pointfixe.m' to compute the approximate solution of f with a precision of $eps = 10^{-2}$, in the interval $[0, 1]$.

Program: 'pointfixe.m'

Inputs	Outputs
$\boldsymbol{f}$: Auxiliary function	alpha: Root obtained by the method
x_0: Initial point	error : Estimated error
itmax : Maximum number of iterations	niter : Number of iterations performed
eps: Given precision (Stopping criterion)	

```
function [alpha, error, niter] = pointFixe(g, x0, itmax, eps)
    alpha = x0;
    for niter = 1 : itmax
        x = alpha;
        alpha = g(x);
        error = abs(alpha - x);
        % Check for convergence
        if error < eps
            return
        end
    end
    % Display a warning if the method did not converge
    warning('The fixed-point method did not converge');
end
```

Exercise 3 :

Find the root of the function $f(x) = \ln x + x - 2$ in the vicinity of $x_0 = 1$, using the function 'newton.m'. Program: 'newton.m'

Inputs	Outputs
$\boldsymbol{f}$: Auxiliary function	alpha: Root obtained by the method
x_0: Initial point	error : Estimated error
itmax : Maximum number of iterations	niter : Number of iterations performed
eps: Given precision (Stopping criterion)	

```
function [root, error, niter] = newtonRaphson(f, df, x0, tol, Itmax)
    % f: the function whose root we are looking for
    % df: the derivative of the function
    % x0: initial estimate of the root
    % tol: tolerance for the stopping criterion
    % maxIter: maximum number of iterations
    % Initialize variables
    root = x0;
    % Iterative process
    for niter = 1:Itmax
        fx = f(root);
        dfx = df(root);
        root = root - fx / dfx;
        % Calculate the error
        error = abs(f(root));
        % Check for convergence
        if error < eps
            return
        end
    end
```