```
function [ x_root ] = Bisection_Method( fx, a, b )
% Bisection Method Function
% fx is the function of x
% a and b are the constants, where x lies between them.

Fa = fx(a); % Determine f(a)
Fb = fx(b); % Determine f(b)

max_Iteration = 50; % Maximum Iteration to stop the funciton
error_tolerance = 1e-6;   % Error tolerance of the root

% If both f(a) and f(b) are both positive or both negative
if Fa*Fb > 0

    % The a and b range is invalid. Therefore, no answer.
    fprintf('a.) Error: The functions have the same sign at points a and b.\n\n')
    x_root = ('No Answer');

else

    for count = 1: max_Iteration

        % Bisection Method
        x_root = (a + b)/2;

        % Calculate error tolerance
        tolerancez = abs((b-a)/2);

        % Calculate f(x)
        FxNS = fx(x_root);

        % If f(x) = 0
        if FxNS == 0

            % x_root is the solution
            fprintf('An exact solution x =%11.6f was found.', x_root)

            % Break for loop.
            break

        % If the error tolerance is less than 1e-6
        elseif tolerancez < error_tolerance

            % Break for loop
            break

        % If the maximum counter reached
        elseif count == max_Iteration

            % Exit program
            fprintf('Solution was not obtained in %i iterations.', max_Iteration)

            % Break for loop
            break

        % If f(a)*f(b) < 0
        elseif fx(a)*FxNS < 0

            % b is the root.
            b = x_root;
        else
```

```
            % a is the root, otherwise
            a = x_root;

        end
    end
end
end
```