General requirements:

1. all root finding functions must have optional outputs with the function value at solution point, and number of iterations. So the general root finding function definition should look like

   \[
   \text{function } [x_{\text{sol}}, f_{\text{at}\ x_{\text{sol}}}, N_{\text{iterations}}] = \text{find\_root\_method}(f_{\text{handle}}, \ldots)
   \]

2. all relevant input parameters should be validated against possible user errors.

   All methods should be tested for the following parameters \( \epsilon_f = 1e^{-8} \) and \( \epsilon_x = 1e^{-10} \). Wherever the initial bracket is not applied (for example Newton-Raphson algorithm) use the right limit of the initial bracket as a starting point of the algorithm.

Problem 1 (5 points)
Write proper implementation of the bisection algorithm. Define your function as

   \[
   \text{function } [x_{\text{sol}}, f_{\text{at}\ x_{\text{sol}}}, N_{\text{iterations}}] = \text{bisection}(f, x_n, x_p, \epsilon_f, \epsilon_x)
   \]

   Test your implementation with \( f(x) = \exp(x) - 5 \) at initial bracket \([0,3]\)

Problem 2 (5 points)
Write proper implementation of the false position algorithm. Define your function as

   \[
   \text{function } [x_{\text{sol}}, f_{\text{at}\ x_{\text{sol}}}, N_{\text{iterations}}] = \text{regula\_falsi}(f, x_n, x_p, \epsilon_f, \epsilon_x)
   \]

   Test your implementation with \( f(x) = \exp(x) - 5 \) at initial bracket \([0,3]\)

Problem 3 (5 points)
Write proper implementation of the secant algorithm. Define your function as

   \[
   \text{function } [x_{\text{sol}}, f_{\text{at}\ x_{\text{sol}}}, N_{\text{iterations}}] = \text{secant}(f, x_1, x_2, \epsilon_f, \epsilon_x)
   \]

   Test your implementation with \( f(x) = \exp(x) - 5 \) at initial bracket \([0,3]\)

Problem 4 (5 points)
Write proper implementation of Newton-Raphson algorithm. Define your function as

   \[
   \text{function } [x_{\text{sol}}, f_{\text{at}\ x_{\text{sol}}}, N_{\text{iterations}}] = \text{NewtonRaphson}(f, x_{\text{start}}, \epsilon_f, \epsilon_x, df_{\text{handle}})
   \]

   Test your implementation with \( f(x) = \exp(x) - 5 \) at initial bracket \([0,3]\)
Problem 5 (5 points)
Write proper implementation of Ridders’ algorithm. Define your function as
\[
\text{function } [x_{\text{sol}}, f_{\text{at } x_{\text{sol}}}, N_{\text{iterations}}] = \text{Ridders}(f, x_1, x_2, \text{eps}_f, \text{eps}_x)
\]
Test your implementation with \( f(x) = \exp(x) - 5 \) at initial bracket \([0,3]\).

Problem 6 (5 points)
For each method find the root of the following two functions
1. \( f_1(x) = \cos(x) - x \) with the 'x' initial bracket \([0,1]\)
2. \( f_2(x) = \tanh(x - \pi) \) with the 'x' initial bracket \([-10,10]\)

Make a comparison table for the above algorithms with following columns
1. Method name
2. root of \( f_1(x) \)
3. initial bracket or starting value used for \( f_1 \)
4. Number of iterations to solve \( f_1 \)
5. root of \( f_2(x) \)
6. initial bracket or starting value used for \( f_2 \)
7. Number of iterations to solve \( f_2 \)

If algorithm diverges with provided initial bracket, appropriately modify the bracket. Indicate modified bracket used in the above table as well. Make your conclusions about speed and robustness of the methods.