Homework 03

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
`function [x_sol, f_at_x_sol, N_iterations] = find_root_method(f_handle,...)`

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

All methods should be tested for the following parameters eps_f=1e-8 and eps_x=1e-10.

Where ever 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
`function [x_sol, f_at_x_sol, N_iterations] =bisection(f, xn, xp, eps_f, eps_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
`function [x_sol, f_at_x_sol, N_iterations] =regula_falsi(f, xn, xp, eps_f, eps_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
`function [x_sol, f_at_x_sol, N_iterations] =secant(f, x1, x2, eps_f, eps_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
`function [x_sol, f_at_x_sol, N_iterations] =NewtonRaphson(f, xstart, eps_f, eps_x, df_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

```
function [x_sol, f_at_x_sol, N_iterations] =Ridders(f, x1, x2, eps_f, 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. $f1(x) = \cos(x) - x$ with the 'x' initial bracket [0,1]
2. $f2(x) = \tanh(x - \pi)$ with the 'x' initial bracket [-10,10]

Make a comparison table for the above algorithms with following columns

1. Method name
2. root of $f1(x)$
3. initial bracket or starting value used for $f1$
4. Number of iterations to solve $f1$
5. root of $f2(x)$
6. initial bracket or starting value used for $f2$
7. Number of iterations to solve $f2$

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