The purpose of this lab is to implement and compare the Newton, and Secant iterative methods.

Develop two procedures NEWTON, and SECANT as follows:

NEWTON implements the Newton-Raphson iterative method. The input parameters must include the starting number for the iteration. Note also that two additional procedures (or functions) RFUNC and RDERIV must be included for the function for Newton iteration and its derivative respectively. These will have to be modified for each example treated.

SECANT implements the Secant iterative method. The input parameters must include two starting numbers for the iteration. Note also that one must also create a procedure (or function) SFUNC, and this will have to be modified for each example treated. You may use RFUNC from the Newton section for this.

For each of the above procedures you must also include a variable TOL for the desired error tolerance and a variable ITERS for the maximum number of allowable iterations. You should calculate both the "relative error" and "absolute error" in terms of the kth and (k+1)st iteration values and compare with TOL for your information, but only use the relative error for satisfying the error criterion.

Write a program which uses these procedures to solve some nonlinear equations. Printed output should consist of the equation being solved, TOL, ITERS, the actual number of iterations to satisfy the error criterion, the starting value, the list of actual successive iterations, and the final solution. Organize your output in an easy to understand and well thought out style. A well documented listing of your source program and procedures should also be included. Be sure to explain any unusual behavior in the examples and compare the methods. Note that you must also choose the proper formulation for your equations and explain the reason for your choice. Note also that you must choose the starting value or values carefully. Be sure to carry and display enough figures to clearly show the accuracy at each step.

A testing strategy must be designed for verification of the procedures as well as for the comparison of the methods. Some ideas on this will be discussed in class. The software design must consider computational efficiency as well as clarity, modularity, and ease of use.
Apply these programs to the following problems and use TOL=1.0e-5 and ITERS=50.

PROBLEM #1
Find the sixth root of 10.

PROBLEM #2
Find the roots of $x + 8 - 2e^{\sqrt{x}} = 0$.

PROBLEM #3
Find the first and 5th positive roots of $x - \tan(2x) = 0$ where $x$ is in radians.