Remember to indent the bodies of your IF-THEN and looping structures.

For the problems below that require you to implement the bisection method you should use a value of 10^{-10} for each of the tolerances. Note, in order to assign a value of 10^{-10} in MATLAB, you should do

```
tol = 1.0e-10
```

If you enter

```
tol = 10.0e-10
```

then your tolerance will be 10^{-9}. Also, you should allow a maximum of 50 iterations.

1) (3 pts) Write your bisection method as a function instead of a script. In addition, use another function to evaluate $f(x)$ rather than hard coding it into your bisection function. Use the syntax given in the notes for your function calling sequence. Hand in a printout of your function.

2) (3 pts) A few homework assignments ago, we used the `integral` function to evaluate definite integrals. In order to use this function, we had to tell MATLAB what definite integral we wanted to compute. To do that, we had to create a MATLAB `.m` file that contained the integrand (See notes from October 8 for more information).

   This idea can be extended to the bisection function. It is possible to send the $f(x)$ function you want to find the root of as an input variable (and it requires almost no changes to the function you wrote in Problem 1).

   Modify your bisection function from Problem 1 so that the function you want to find the root of is sent in as an input similar to the `integral` function. HINT: The first line of your bisection function should look like

   ```
   function [.....] = bisec(myfun,a,b,tol1,tol2,maxits)
   ```

   You would call this from the main MATLAB workspace using something like

   ```
   [....] = bisec(@Problema,0,1,1.0e-10,1.0e-10,50);
   ```

   Hand out a printout of the modified function.

3) (6 pts) Write a *single* MATLAB script file that runs all three test problems from HW 17. For each case, your script print out the value of the root, the value of the $f(x)$ function at the root, the status indicator, the iteration count and a graph of the two convergence histories (the relative distance and absolute value of the function). You will also need to create a `.m` file that computes the $f(x)$ function for each test case. Hand in printouts of your main script, your function `.m` files, the 3 graphs and the output from running the script.