Computer Exercise 3.1.4

This program will attempt to solve $6(e^x - x) = 6 + 3x^2 + 2x^3$ on the interval $I = [-1, 1]$ using the bisection algorithm. There will be a set maximum amount of iterations to prevent the program from blowing up my computer just in case.

Define $f(x) = 6 + 3x^2 + 2x^3 - 6(e^x - x)$; here, the endpoints of I yield positive and negative pairs as is required to initiate the bisection algorithm.

```
f=@(x) 6 + 3*(x^2) + 2*(x^3) - 6*(exp(x) - x); %define function to use for bisection method
```

```
% attempt on interval I

a=-1;
b=1;
n=10;
error = 10^(-2);

[root1, i] = bisect(f, a, b, n, error);

fprintf(['Calculated root from bisection after %d iterations for the interval [-1,1] ' ...
    'is r = %f which evaluates to \n f(r) = %f; ' ...
    'this yields an error of %f'], i-1, root1, f(root1), abs(f(root1)))
```

```
Calculated root from bisection after 0 iterations for the interval [-1,1] is r = 0.000000 which evaluates to
 f(r) = 0.000000; this yields an error of 0.000000
```

This makes sense because the first evaluation of "c" in the algorithm yields

$c = \frac{a+b}{2} = \frac{-1+1}{2} = 0$, so $f(0) = 6 + 0^2 + 0^3 - 6(e^0 - 0) = 6 - 6 = 0$.

What we have observed here is a very rare occurence in root finding attempts.

```
function [c, i] = bisect(f, a, b, n, error) %I added an extra output to track iteration count
c=(a+b)/2;
i=1; %set iteration counter
%error = |f(c) - 0|
    while (abs(f(c))>error) && (i<=n) %exit while loop once error tolerance
       %  or max iterations has been reached
        if f(a)<0 && f(c)<0 %if true, then f(a)f(c)>0
            a=c;
        else % otherwise, f(b)f(c)>0
            b=c;
        end
        c=(a+b)/2;
        i = i+1; %update iteration counter
    end
    if (abs(f(c))>error) %if this triggers, this means that max iterations
       %  has been reached before error tolerance
        fprintf(['bisection algorithm was unsucessful after %d iterations; ' ...
```

```
            'error = %f'], i-1, abs(f(c)))
    end
end
```