ECE 3340 – Programming Assignment 1
Finding Roots of Equations

This assignment is designed to teach you how to solve two of the most common problems in numerical methods: (1) evaluating polynomial functions and (2) finding roots of those functions.

The problem
Your application will take a series of command-line arguments $[c_n, c_{n-1}, c_{n-2}, \cdots, c_1, c_0, a, b]$ as floating point values. The values c_0 through c_n will be the coefficients of a degree n polynomial $p(x)$. The coefficients are specified in the order of decreasing power. For example, c_0 is a constant and c_n is the coefficient of x^n. The last two values, a and b, will specify the interval over which your application will search for a root and return the following:

a) The location $x = r_0$ of the root according to the **bisection method**
b) The number of iterations required to locate the root using the bisection method
c) The location $x = r_1$ of the root according to **Newton's method**
d) The number of iterations required to locate the root using Newton's method
e) The value of $f(a)$, $f(b)$, $f(r_0)$, $f(r_1)$, and $f'(r_1)$

Your code will operate under the following constraints:

1) Evaluate both methods to machine precision (describe how you test this in the comments)
2) For Newton's method, start with $x_0 = \frac{a+b}{2}$
3) Return an error if no root is found within the interval $[a, b]$

Structure your code so that you have three independent functions for calculating:

- the bisection method
- Newton's method
- Horner's algorithm

When writing a function for Horner's algorithm, remember that you'll need to make it flexible enough to solve both the value of the polynomial and its derivative, when necessary.

Turning in your code
Your code must be written in C or C++ and uploaded to Blackboard as a single text file. I will download it and use a script to test your code out for several examples, so don't do anything weird with the output. Output the values in the order (a through e) described above as standard console output (use `std::cout` or `printf`). You can label the values with text, just don't add any other numbers or change their order. Also, please don't collect any data via the console (everything must be passed via command line parameters).

I will be compiling and grading the code using Visual Studio 2012 (VC++ 11.0). This is both the industrial standard and consistent with the rest of the department. Visual Studio 2012 is available in the campus computer labs and through Microsoft Dreamspark, where students can obtain the full (professional) version. I certainly don't mind if you write the code on another platform, but make sure to test it using Visual Studio 2012.

Examples

$x^3 - 6x^2 + 4x + 12 = 0$	$[1, 4]$	$x_r = 2.5173$	args: 1 -6 4 12 1 4
$x^3 - 6x^2 + 4x + 12 = 0$	$[4, 6]$	$x_r = 4.53407$	args: 1 -6 4 12 4 6
$x^6 - 14x^4 + 49x^2 - 36 = 0$	$[1.5, 2.5]$	$x_r = 2$	args: 1 0 -14 0 49 0 -36 1.5 2.5
$x^7 - 14x^5 + 49x^3 - 36x = 0$	$[-4, -2.5]$	$x_r = -3$	args: 1 0 -14 0 49 0 -36 0 -4 -2.5

ECE 2331 – Programming Assignment 1
Finding Roots of Equations
Rubric

Name_____________________________

input format		__________ **/ 5**
output format		__________ **/ 5**
Horner's algorithm		__________ **/ 15**
appropriate precision	__________ / 5	
code is correct / robust	__________ / 5	
input and code commented	__________ / 5	
Bisection method		__________ **/ 35**
correct number of iterations	__________ / 10	
correct root	__________ / 10	
fails robustly	__________ / 5	
input and code commented	__________ / 10	
Newton's Method		__________ **/ 40**
correct number of iterations	__________ / 10	
correct root	__________ / 20	
input and code commented	__________ / 10	
Total		__________ **/ 100**