This lab investigates doing a binary search to find an approximation to a root of a polynomial.

1. Make a copy of the sample program cubic.cpp and compile and execute it. This program will accept the four coefficients (as double's) of a cubic polynomial (e.g. \( p(x) = ax^3 + bx^2 + cx + d \) with \( a \neq 0 \)) and it will find an interval \([\text{left}, \text{right}]\) where the polynomial changes sign. You may want to put the four coefficients in a file cubic.txt and execute the program via

   \texttt{cubic < cubic.txt <cr>}

2. Examine the code and note that the coefficients are global variables and that the values of the polynomial are computed by the function

   \texttt{double cubic(double x);}

   Given an interval where a polynomial \( p(x) \) changes sign how do you know that there is at least one root in the interval?

3. We probably will not be able to find the \textit{exact} value of the root but we could find the \textit{approximate} value with the following binary search algorithm. Suppose that the polynomial changes sign in \([\text{left}, \text{right}]\). We could compute the value of the polynomial at the \textit{midpoint} and if the polynomial changes sign in \([\text{left}, \text{midpoint}]\) we could set

   \texttt{right = midpoint;}

   otherwise if the polynomial changes sign in \([\text{midpoint}, \text{right}]\) we could set

   \texttt{left = midpoint;}

   the only other possibility is that the polynomial is zero at the midpoint in which case we have found a root! You should see that you could write a procedure

   \texttt{void binary_search(double & left, double & right);} to do this.

3. Suppose you want to find the approximate value of the root to within 0.00001. How would you do this with a loop which checks the remaining distance between \texttt{right} and \texttt{left}?

   \textbf{Assignment} Write a program which will be an addition to the sample program cubic.cpp so that the new program implements a binary search for the root accurate to within 0.00001. Your program \textbf{must include} the following function:

   \texttt{void binary_search(double & left, double & right);} which will accept \texttt{left} and \texttt{right} passed by \textit{reference} so one of the endpoints can be updated by the \textit{midpoint}. You should include some diagnostics which print out the successive intervals \([\text{left}, \text{right}]\) as they narrow in on the root. Your program should also print out the final approximation to the root.

   Email me the pathname of your program \textit{in plain text, not as an attachment}. For example, you might say

   \texttt{My lab4 program is /usr/stu/demo/cs221/lab4.cpp}