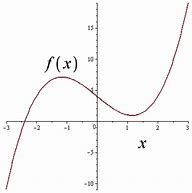
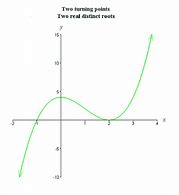
**ENGR 102**

**Lab 6A [100 Points]**

**Root Finding – To be done as a team**

As a team, you are to write a program that will find a root of a cubic polynomial. A cubic polynomial is of the form:

A root of the polynomial is a value, x, such that *f(x)*=0. For a generic cubic polynomial, there will be one local maximum, and one local minimum, and the curve will go off to negative infinity in one direction, and positive infinity in another. Here are three examples:

[](https://www.bing.com/images/search?view=detailV2&ccid=QSL%2bJQG8&id=6C32DFB6D7D8B09F5E16A834BBB8C44C6243F9B5&thid=OIP.QSL-JQG8vggP1OHYw2xMKwHaHa&mediaurl=https://upload.wikimedia.org/wikipedia/commons/thumb/a/a3/Polynomialdeg3.svg/1200px-Polynomialdeg3.svg.png&exph=1200&expw=1200&q=cubic+polynomial&simid=608020569307676849&selectedIndex=7&qpvt=cubic+polynomial)[](https://www.bing.com/images/search?view=detailV2&ccid=Ms62Top8&id=6F99D300A3BFA7A4F2A507598E3CED1E513D6429&thid=OIP.Ms62Top8zE8DALOO8ApNyQHaHa&mediaurl=http://www.thephysicsmill.com/blog/wp-content/uploads/cubic_figure_1_root.png&exph=400&expw=400&q=cubic+polynomial&simid=608027174969675359&selectedIndex=48&qpvt=cubic+polynomial)[](https://www.bing.com/images/search?view=detailV2&ccid=tGlju7Om&id=413468A61343B04643903535EB1CD64AD625B0E9&thid=OIP.tGlju7Om6axJczRGcj0jpgHaIG&mediaurl=http://www.biology.arizona.edu/biomath/tutorials/polynomial/graphics/polynomial_cubic3.gif&exph=520&expw=475&q=cubic+polynomial&simid=608013547131896831&selectedIndex=4&qpvt=cubic+polynomial)

The polynomial has some number of real roots: the points at which the curve crosses the x-axis. A cubic curve will have either 3 real roots (as above at left), 1 root (above, middle), or in rare cases 2 roots (as above at right). Note that roots are typically single roots, and for single roots, the curve is negative on one side of the root, positive on the other (a double root results in a tangent to the curve, like that above at right).

You should write a program that requests a user for the following input: 1) coefficients of the polynomial: A, B, C, and D, and 2) bounds on one single root: *a*, *b*. The user *should be expected* to input *a* and *b* such that a<b and there is exactly one single root of the polynomial between *a* and *b*. You should report the value of the root, accurate to within 10-6.

*Before developing your code, your team should come up with at least 4 test cases. Test cases should indicate a curve (the 4 coefficients for the curve), as well as a large starting bound on the root (values of a and b, with b >= a+1). Your code should include documentation of these 4 (or more) test cases in comments, near the top of the program (after your standard header).* **[10 Points]**

**To solve this problem, you can use the following procedure, known as bisection.** It is a form of what is more generally known as binary search:

* Because the root is a single root, either *f(a)* or *f(b)* will be positive, and the other will be negative.
* You can generate the midpoint of the interval from *a* to *b*, by finding the value halfway between *a* and *b*. Call this point *p*.
* If you evaluate *f(p)*, it will be either positive or negative (or possibly, be the root itself, if *f(p)*=0).
* This will enable you to narrow the interval to either [*a*,*p*] or to [*p*,*b*].
* You can continue doing this until your interval is less than 10-6 wide.

As a team, you should create a program that performs bisection to determine the root. More specifically, your program should do the following:

1. Read in the coefficients of the polynomial from the user **[10 Points]**
2. Read in the upper and lower bounds around a single root of the polynomial **[10 Points]**
3. Determine the value of that root to within 10-6 **[50 Points]**
4. Print the result of that root finding, as a single number **[10 Points]**
5. Print out how many iterations it took to find that root. **[10 Points]**

Be sure to include comments in your code.

Challenge: Cubic polynomials always have one root that can be found through bisection. There also are ways to find maximum and minimum bounds on the sizes of real roots of polynomials (you will need to research this). Modify your program so that it takes in only the coefficients of the polynomial, and reports one of the real roots to within 10-6.