

# Assignment 1

In this problem you will create a Python module and also two scripts that use the module.

## Polynomial Module

The Python module will provide two functions that work with polynomials. A polynomial will be represented by a list of the coefficients, from lowest degree to highest. So, for example, the polynomial

$$2 + 3 \cdot x + 4 \cdot x^2 + 7 \cdot x^4$$

would be represented by the list

[2, 3, 4, 0, 7]

Note the 0 for the missing term  $x^3$ .

The module will provide two functions: one to evaluate a polynomial at a given value of  $x$ , another to find a root of a polynomial using the bisection method.

Use the following as the headers for the two methods:

```
def eval(x, poly):
    """
    Evaluate the polynomial at the value x.
    poly is a list of coefficients from lowest to highest.

    :param x:      Argument at which to evaluate
    :param poly:   The polynomial coefficients, lowest order to highest
    :return:       The result of evaluating the polynomial at x
    """

def bisection(a, b, poly, tolerance):
    """
    Assume that poly(a) <= 0 and poly(b) >= 0.

    :param a: poly(a) <= 0  Raises an exception if not true
    :param b: poly(b) >= 0  Raises an exception if not true
    :param poly: polynomial coefficients, low order first
    :param tolerance: greater than 0
    :return:  a value between a and b that is within tolerance of a root of the polynomial
    """
```

One way to implement `bisection` is this: as long as the the absolute value of the difference of `a` and `b` is greater than `tolerance` do this: compute `mid` as the average of `a` and `b`; evaluate the polynomial with `mid` as the argument; if the polynomial evaluates to negative, assign `mid` to `a`; otherwise, assign `mid` to `b`.

## Script 1

The first script should begin by defining two variables, a value for  $x$  and a polynomial. Use `eval` to evaluate the polynomial at the argument value and print the result.

## Script 2

The second script should begin by defining 4 variables: a value for `a`; a value for `b`; a polynomial list; a value for `tolerance`. The script should use `bisection` to determine a root for the polynomial. Print the value of the root and also the result of evaluating the polynomial at that root.

## Testing

The following script will test your module:

```
from polynomials import bisection, eval

toler = 1e-14

poly1 = [-945, 1689, -950, 230, -25, 1]
# roots are 1, 3, 5, 7, 9
x1 = bisection(0, 2, poly1, 1e-15)
# print root and evaluate the polynomial
print(x1, eval(x1, poly1))
x2 = bisection(4, 2, poly1, toler)
print(x2, eval(x2, poly1))
x3 = bisection(4, 6, poly1, toler)
print(x3, eval(x3, poly1))
x4 = bisection(8, 6, poly1, toler)
```

```
print(x4, eval(x4,poly1))
x5 = bisection(8,100,poly1, toler)
print(x5, eval(x5,poly1))
# compare the roots to the expected values
print(x1-1, x2-3, x3-5, x4-7, x5-9)
```

## Submitting the Assignment

Please archive the entire project and submit using the dropbox in D2L.

## Due on June 4, 2015

## Links

- [Home](#)
- [Assignments](#)
- [Extra](#)
- [Notes](#)
- [Resources](#)
- [Syllabus](#)
- [Timetable](#)