



## CSCI 310 – 02 (Fall 2019)

### Programming Foundations

#### Programming Assignment 2

DUE: Wednesday, Sep 25, 11:59 PM (turnin time)

### Overview

This is essentially the same as the previous programming assignment with some minor modifications. The biggest modification is the use of a *linked list* to represent a polynomial.

### Specifications

In addition to updating our representation of a polynomial from using an array to using a linked list, in this program in addition to being able to evaluate the values of a polynomial, say

$$f(x) = a_0x^n + a_1x^{n-1} + \cdots + a_{n-1}x^1 + a_nx^0$$

your class definition can also evaluate the polynomial's *derivative*. The derivative of a polynomial of degree  $n$  is defined as the following polynomial of degree  $n - 1$ :

$$\frac{df(x)}{dx} = f'(x) = a_0nx^{n-1} + a_1(n-1)x^{n-2} + \cdots + a_{n-2}2x^1 + a_{n-1}x^0.$$

All input and output requirements remain as in the previous assignment.

### Input

Your program should accept an even number of lines of text. Each pair of lines will represent one problem. The first line will contain a list of  $n + 1$  integers  $a_n, a_{n-1}, \dots, a_1, a_0$  which represent a set of coefficients to a polynomial expression. The degree of the polynomial is  $n$ . The coefficients are paired with the terms of the polynomial in the following manner:

$$a_nx^n + a_{n-1}x^{n-1} + \cdots + a_1x^1 + a_0x^0$$

The second line of text are the  $m + 1$  values  $x_0, x_1, \dots, x_m$  to be used for  $x$  when evaluating the *derivative of the polynomial*.

Input will be redirected from standard input via redirection on the \*nix command prompt. As in the previous assignment, feel free to use the `split()` function provided at: <http://www.ecst.csuchico.edu/~bjuliano/csci310/Code/split.cpp>.

### Sample Input

```
-2
5 0 1 6
1 -1
7 6 -1
-1 1
7 6 -1
3 0 0 -4 0 2
1 2 3 4 5
```

How many polynomials are provided in the input above? What are they? What are their derivatives?

### Output

For each pair of lines, your program should evaluate the polynomial for all the  $m + 1$  values of  $x$  ( $x_0$  through  $x_m$ ) and output the resulting values on a single line.

Output should be sent to standard output and must exactly follow the format in the sample below.

### Sample Output

```
0 0 0 0
1 1 1
-1 -1 -1
7 224 1191 3808 9335
```

## Additional Requirements

You are required to design a `LinkedPolynomial` class that inherits from a provided `PolynomialInterface` abstract class. This `LinkedPolynomial` class will use a *linked list* to represent a polynomial. You are required to use the `Node` class provided in our textbook. Refer to the linked list implementation of the Bag ADT provided in our textbook (see `LinkedBag.h`) as a reference on how you will implement your `LinkedPolynomial` class. Your driver program does not have to use all the member functions of your `LinkedPolynomial` class to solve the problem given. However, you may choose to use the `print()` member function that displays a polynomial object when debugging your code.

There is no requirement on how you maintain the terms in a polynomial each time the member function `addTerm()` is called. For simplicity, you can just add the new term at the start of the linked list, just like with the `LinkedBag` in our textbook.

Your submission will consist of the following files, submitted using the Department of Computer Science's `turnin` facility:

- `LinkedPolynomial.h` – specification/header file for `LinkedPolynomial` class
- `LinkedPolynomial.cpp` – implementation file for `LinkedPolynomial` class
- `evalPoly.cpp` – driver code containing `main()` function

Note that the code for the `PolynomialInterface` abstract class will already be available on `turnin`. You should not edit nor deviate from the specifications given in this interface (abstract class).

We want you to develop good code documentation habits. Source code solutions submitted without any meaningful documentation will receive a total score of zero (0). You may refer to the *Google C++ Style Guide* section on source code comments as a guide.

Be sure to also review and adhere to the **Coding Standards** for this course.