

# CS 231- Programming Language Design

## Linked List Data Structure for Multiplying Polynomials

### Requirements:

In this programming assignment, you will demonstrate your knowledge of linked lists by writing an application using linked lists. You will write a data structure package (a separate collection of functions in .c file) that implements linked list data structure.

You will use this package in an application that reads pairs of polynomials and multiplies each pair of polynomials and displays results.

For the data structure part of implementing linked list data structure, you may use any variation of linked list. You may use a singly linked list with head only, a singly linked list with head and tail, or a doubly linked list.

The following functions should be included in the linked list package. You should decide the parameters and return type of the functions. Note that only required parameters should be sent to each function

`initList`- initializes the list  
`isEmpty` -checks if the linked list is empty  
`insertFirst`- adds a given new element to the head of the list  
`insertLast`- adds a given new element to the end of the list  
`insertAfter`- inserts a given element after the first occurrence of another given element.  
`insertSorted`-given an element to insert, inserts the element so that the list is in increasing order of the data.  
`deleteFirst`- deletes the element at the head  
`deleteLast` -deletes the element at the end of the list  
`replaceElement`- replaces every occurrence of given element with the new element  
`deleteElement`-deletes the first occurrence of element if it is there  
`searchElement`- searches for a given element and returns the node of the linked list containing the element or returns null if it not there  
`containsElement`-returns 1 if the element is there, otherwise returns 0 .

toString: returns the string format of all elements of the list

The list of polynomials used for multiplication will be read from a file whose name will be given in the command line. The file contains a pair of polynomials per line of input. The character \* (star) separates the polynomials in the pair. Each polynomial is presented as a sequence of terms. The terms are separated by character “;”(semicolon). Each term consists of *co-efficient and an exponent*, which are separated by “,”(comma). Note that co-efficient and exponent pair may appear in any order. Same exponent might appear in multiple terms. The coefficient may be positive, negative or 0. The exponent will be 0 or positive integer. An example of a line from the input is shown below.

3,2;3,4;2,1\* -2,2;2,1;1,0;1,1

You will need to multiply the polynomials and display results to the standard output. The display of the product polynomial must contain only one term per each exponent and it must have exponents in the decreasing order.

3,2;3,4;2,1\* -2,2;2,1;1,0;1,1

$-4, 3; 3, 4 \quad * -2, 2; 3, 1; 1, 3$

The first line then contains the following polynomials:

$3x^4 + 3x^2 + 2x^1$  and  $-2x^2 + 3x^1 + 1x^0$

After reading the file, provide the following output:

$3x^4 + 3x^2 + 2x^1$  times  $-2x^2 + 3x^1 + 1x^0$  is  $-6x^6 + 9x^5 - 3x^4 + 5x^3 + 9x^2 + 2x^1 + 1$   
 $3x^4 - 4x^3$  times  $x^3 - 2x^2 + 3x^1$  is  $3x^7 - 10x^6 + x^5 - 12x^4$

### **Internal Requirements:**

1. As always, your program should use good style, as defined by the Style Guide.
2. The linked list package cannot do any I/O. The linked list can be returned as a string and then printed from the application program.
3. No recursion except in the definition of the linked list.
4. When you print the exponent, you may print it using a ^ rather than as a superscript.