## Linked List (21 pts)

1. (3 pts) Return the size of the linked list
2. (3 pts) Print the linked list
3. (3 pts) Test if a value x is contained in the linked list
4. (3 pts) Add a value x if it is not already in the linked list (add at the end of the list)
5. (3 pts) Remove a value x if it is contained in the list
6. (3 pts) Reverse the order of the items in the linked list
7. (3 pts) Given two lists, L1 and L2, create a new list L3 that contains the intersection (common data elements) of L1 and L2.

## Polynomial Operations Using Linked Lists (30 pts)

1. **Input** - Enter the position in the poly array to store this polynomial and the number of terms in the polynomial. Then, perform a loop for the number of terms, entering coefficient and exponent pairs. To enter the polynomial 5.1x4 + 6x3 − x + 8 (4 terms) in position 0 of the poly array we would enter at the terminal: (sample menu of choices for user)

Please enter what you want:

i for input

a for add

s for subtract

m for multiply

e for evaluate

p for print

q for quit

i

input: enter index number of polynomial and how many terms

0 4

enter coef and exponent for term 1

5.1 4

enter coef and exponent for term 2

6 3

enter coef and exponent for term 3

-1 1

enter coef and exponent for term 4

8 0

5.1x^4 + 6.0x^3 + -1.0x^1 + 8.0x^0

1. **addPoly(LinkedList L1, LinkedList L2, LinkedList L3)**
2. **subtractPoly(LinkedList L1, LinkedList L2, LinkedList L3)**
3. **multPoly(LinkedList L1, LinkedList L2, LinkedList L3)**
4. **evalPoly(LinkedList L1, int value)**
5. **printpoly(L1)**