



# Data Structures and Algorithms

## Lab 2 – Recursion & Linked List

With the following struct:

```
struct node{  
    int data;  
    node* next = NULL;  
};
```

Complete question 1 to 5

### Question 1

Write a function to convert a list to a linked list.

For example:

```
// Given a list with some values (data) in it  
int List[5] = { 1, 2, 3, 4, 5};  
int Size = 5;  
  
// This function will retrieve all data from list  
// and create a linked list from that data  
// For example:  
node* linkedList = ConvertToLinkedList(List, Size); //1->2->3->4->5  
PrintLinkedList(linkedList); // return 1 2 3 4 5
```

### Question 2

Write a program having these following functions:

- Traversal a linked list and print all data.
- Sort the linked list.

For example:

```
// Initialize data  
int List[5] = { 5, 8, 3, 2, 9 };  
int Size = 5;  
node* LinkedList = ConvertToLinkedList(List, Size);
```



```
// Question 2.a - Print all data
PrintLinkedList(LinkedList); // return 5 8 3 2 9

// Question 2.b - Sort a linked list
SortLinkedList(LinkedList, Size);

PrintLinkedList(LinkedList); // return 2 3 5 8 9
```

### Question 3

Write a function to reverse a linked list recursively.

For example:

```
PrintLinkedList(head); // 1 -> 5 -> 7 -> 9 -> 2
node* newHead = ReverseList(head);
PrintLinkedList(newHead); // 2 -> 9 -> 7 -> 5 -> 1
```

Hint:

You can use these prototypes:

```
node* reverseList(node* pHead);

node* rvList(node* previousNode, node* currentNode);
```

Use following data structure to solve question 4, 5, 6

A polynomial can be represented by a linked list as following:

$$1 + 5x + 7x^2 + 9x^3 \Leftrightarrow 1 \rightarrow 5 \rightarrow 7 \rightarrow 9$$
$$-2 + 10x^4 \Leftrightarrow -2 \rightarrow 0 \rightarrow 0 \rightarrow 0 \rightarrow 10$$

### Question 4

Write a function to add (or subtract) two polynomials.

For example:

```
// Initialize data
int Poly1[5] = { 5, 8, 3, 2, 9 };
int Poly2[5] = { 2, 0, 0, 1, 5 };
node* PolyList1 = ConvertToLinkedList(Poly1, 5);
node* PolyList2 = ConvertToLinkedList(Poly2, 5);
```



```
// Question 4
node* addedPoly = AddPoly(PolyList1, PolyList2);
PrintLinkedList(addedPoly); // return 7 8 3 3 14
```

## Question 5

Write a function to multiply two polynomials.

```
// Initialize data
int Poly1[5] = { 5, 8, 3, 2, 9 };
int Poly2[5] = { 2, 0, 0, 1, 5 };
node* PolyList1 = ConvertToLinkedList(Poly1, 5);
node* PolyList2 = ConvertToLinkedList(Poly2, 5);

// Question 5
node* mulPoly = MulPoly(PolyList1, PolyList2);
PrintLinkedList(mulPoly); // return 10 16 6 9 51 43 17 19 45
```