CS1 LAB - Recursion with Linked List

Now, you already know about linked list and also recursion. Now in this lab we will try to write code on sorted linked list insertion and deletion using recursion. We will also write a set of other utility functions using recursion!

Here is the list of functions and you **have to use recursion for all of them**:

- struct node* insert(struct node* list, int d);
- struct node* del(struct node* list ,int d);
- void print(struct node *list);
- void freeList(struct node* list);
- void copy (struct node *q, struct node **s);

For your help, the prototypes and the main function is provided here:

```
int main( ) {

    int number = 0, choice=0;
    struct node *pList=NULL;
    struct node *nList = NULL;

    // Let the user add values until they enter -1.
    while(choice!= 4)
    {
        // Get the operation.
        printf("\nDo you want to (1)insert, (2)delete, (3)Copy (4)quit.\n");
        scanf("%d", &choice);

        printf("Your choice is %d\n", choice);

        // Execute the operation.
        if (choice == 1)
        {
           // Get the number.
            printf("Enter the value to insert\n");
            scanf("%d", &number);
            pList = insert(pList, number);
                    // Look at the list.
            printf("Items in linked list: ");
            print(pList);
            //printf("\n");
        }
        else if (choice == 2)
        {    // Get the number.
            printf("Enter the value to delete.\n");
            scanf("%d", &number);
            pList = del(pList, number);
                    // Look at the list.
            printf("Items in linked list: ");
            print(pList);
            //printf("\n");
        }
        else if (choice == 3)
        {
            if (nList)
                freeList(nList);

            copy(pList, &nList); //passing reference of nList as it is not
returning anything
                    // Look at the list.
            printf("Items in NEW linked list: ");
            print(nList);
           // printf("\n");


        }
        else
        {
              break;

        }

    }
    freeList(nList);
    freeList(pList);
    printf("\nBye..\n");
    return 0;
}
```

Most of the functions will be solved in the lab and it is expected that you will follow the TA and will try to work on it and submit your work in webcourses under Lab7. The code written in the lab will not be uploaded as you are typing it anyway and you will be adding more feature.

Also, use memory leak detector to check whether you have memory leak or not! It should be also part of your submission.

Sample input/output:

```
(starts with a new line)
Do you want to (1)insert, (2)delete, (3)Copy (4)quit.
1
Your choice is 1
Enter the value to insert
4
Items in linked list: ->4
Do you want to (1)insert, (2)delete, (3)Copy (4)quit.
1
Your choice is 1
Enter the value to insert
```

```
6
Items in linked list: ->4->6
Do you want to (1)insert, (2)delete, (3)Copy (4)quit.
1
Your choice is 1
Enter the value to insert
2
Items in linked list: ->2->4->6
Do you want to (1)insert, (2)delete, (3)Copy (4)quit.
1
Your choice is 1
Enter the value to insert
1
Items in linked list: ->1->2->4->6
Do you want to (1)insert, (2)delete, (3)Copy (4)quit.
1
Your choice is 1
Enter the value to insert
3
Items in linked list: ->1->2->3->4->6
Do you want to (1)insert, (2)delete, (3)Copy (4)quit.
2
Your choice is 2
Enter the value to delete.
3
Items in linked list: ->1->2->4->6
Do you want to (1)insert, (2)delete, (3)Copy (4)quit.
3
Your choice is 3
Items in NEW linked list: ->1->2->4->6
Do you want to (1)insert, (2)delete, (3)Copy (4)quit.
4
Your choice is 4

Bye..
```