LINKED LIST - II

Counting number of nodes in a list
Printing a list
Searching for a specific node in a list
Creating a list
Reversing a list
Deleting a specific node
Inserting a node in a sorted list

The following site gives a good coverage of linked lists.
http://cslibrary.stanford.edu/103/LinkedListBasics.pdf

Counting the nodes in a List

- *Recursive version:*

```
int count (struct node * pHead)
{
      if (pHead==NULL)
            return 0;
      else
      return(1 + count(pHead->next));
}
```

- ***Iterative version:***

```
int count(struct node *pHead)
{
      struct node * p;
      int c = 0;

      p = pHead;
      while (p != NULL){
            c = c + 1;
            p = p->next;
      }
      return c;
}
```

Printing the contents of a list

To print the contents, traverse the list from the head node to the last node.

```
int PrintList( struct node *list)
{
 struct node current =   list ;
```

The dummy name *current*" is assigned to the first node of the list . We shall be moving down the list by following the *next* link of *current* , without disturbing any item in the list including its name.

```
while (current != NULL)
      {
          printf("%d", current -> data);
          current    =   current -> next;
       }
 return 1;
}
```

Searching for a node containing a specific item

Given a target value, the search attempts to locate the requested node in the linked list. If a node in the list matches the target value, the search returns 1; otherwise it returns 0.

```
// Start with a dummy pointer to headnode
pCur = pHead;
// Search until target is found or we reach
// the end of list
while (pCur != NULL ){
if( pCur->data == target)
return 1;
pCur = pCur->next;
}

// target not found

return 0;
```