```
/* Singly Linked List */
#include <stdio.h>
#include <alloc.h>
#include <stdlib.h>

struct  node
{
int data;
struct node *link;
};

struct  list
{
int count;
struct node *pos;
struct node *head;
}*pList;

struct node *pPrev,*pLoc;

int searchNode(int target)
{

pPrev=NULL;

pLoc=pList->head;

while(pLoc!=NULL && target > pLoc->data)
{
     pPrev=pLoc;
     pLoc=(pLoc)->link;
}

if (pLoc==NULL)
        return 0; /*Not found*/
else
        if (target == pLoc->data) return 1; /*FOUND*/
else
        return 0;
}
```

```
void printList( )
{
int i;
if (pList->count==0) printf("The List is Empty \n");
else
{
        pList->pos=pList->head;
        printf("..... The List data is as follows .....\n");
        for(i=0;i< pList->count;i++)
        {
                printf("%d\t",pList->pos->data);
                pList->pos=pList->pos->link;
        }
        printf("\n******** END OF LIST ************\n");
}
}
```

```
void deleteNode()
{
if (pPrev==NULL)
                        pList->head=pLoc->link;

else                    pPrev->link=pLoc->link;

pList->count =pList->count - 1;
free(pLoc);
}
```

```
void removeNode(int key)
{
int found;

found=searchNode(key);

if (found==1)
        deleteNode();
else
        printf("Error: No matching data found\n");
}
```

```
int insertNode( int dataIn)
{
struct node *pNew;
pNew = (struct node *) malloc(sizeof(struct node));
if (pNew != NULL)
{
        pNew->data=dataIn;

        if (pPrev!=NULL)
        {
                pNew->link=pPrev->link;
                pPrev->link=pNew;
        }
        else
        {
                pNew->link=pList->head;
                pList->head=pNew;
        }
        pList->count+=1;
        return 1;
}
else
        return 0;
}
```

```
void addNode( int dataIn)
{
int found,success;

found = searchNode(dataIn);
if (found==1)printf("Data already inserted\n");
else
{       success=insertNode(dataIn);
        if (success==1)       printf("Data Inserted Successfully\n");
        else    printf("Out of Memory... \n");
}
}
```

```
int menu()
{
int choice;
printf("\n\n********************\n\n");
printf(" ....   M E N U ...\n");
printf("1: Add new data\n");
printf("2: Delete data\n");
printf("3: Print List\n");
printf("4: Quit\n\n");
printf("********************\n\n");

printf("feed in your choice: ");
scanf("%d",&choice);

return choice;
}
```

```
void createList()
{
pList = (struct list *)malloc(sizeof(struct list));
if (pList != NULL)
{
        pList -> head=NULL;
        pList -> count=0;
}
else
{
        printf("Insufficient Memory to create Head Node...Exiting..\n");
        exit(1);
}
}
```

```
void main( )
{
int choice;
int dataIn,deleteKey;

createList();

do
{
        choice = menu();

        if (choice==1)
    {
                printf("Feed in the data: ");
                scanf("%d",&dataIn);
                addNode(dataIn);
    }
        else
        if (choice==2)
        {
                  printf("Enter key to be deleted: ");
                  scanf("%d",&deleteKey);
                  removeNode(deleteKey);
        }
        else
        if (choice==3)
        {
           printList();
        }

} while(choice!=4);

}
```