

Program



```

/* Singly Linked List (insert end, delete begin)*/
#include <stdio.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;

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( )
{

if (pList->count!=0)
{
        pPrev=NULL;
        pLoc=pList->head;
        deleteNode();
}
else
        printf("Error: No data\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 i,succes;
    pPrev=NULL;
    pLoc=pList->head;
    for(i=1;i<=pList->count;i++)
    {
        pPrev=pLoc;
        pLoc=pLoc->link;
    }
    succes=insertNode(dataIn);
    if (succes==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 end\n");
printf("2: Delete begin\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)
    {
        removeNode();
    }
    else
    if (choice==3)
    {
        printList( );
    }
} while(choice!=4);
}
```

HOME OF EDUCATION
Navlaksi®



www.navlaksi.com
Home of Education

DATA Structures@ Navlaksi™

**The
BEST
Teaching** **Superts**

No 1. in Engineering Coaching