



```
#include <stdio.h>
#include <stdlib.h>

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

struct list
{
int count;
struct node *pos;
struct node *head;
}*pList1, *pList2, *pList3;

struct node *pPrev;

void printList(struct list *pList)
{
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");
}
}
```



```
int insertNode( struct list *pList,int dataIn)
{
    struct node *pNew;
    pNew = (struct node *) malloc(sizeof(struct node));
    if (pNew != NULL)
    {
        pNew->data=dataIn;
        pNew->link=pList->head;
        pList->head=pNew;
        pList->count+=1;
        return 1;
    }
    else
        return 0;
}

void addNode(struct list *pList, int dataIn)
{
    int success;
    pPrev='\0';
    success=insertNode(pList,dataIn);
    if (success) printf("Data Inserted Successfully\n");
    else printf("Out of Memory... \n");
}

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

```
void split(int m, int n)
{
    struct node *ploc;
    int count=0;
    ploc=pList3->head;

    while(ploc!=NULL && count<n-m)
    {
        addNode(pList2,ploc->data);
        count++;
        ploc=ploc->link;
    }

    while(ploc!=NULL)
    {
        addNode(pList1,ploc->data);
        ploc=ploc->link;
    }
}
```





```
void main( )
{
int m,n,i;
int dataIn,deleteKey;
pList1=createList();
pList2=createList();
pList3=createList();

printf("enter the number of terms of list: ");
scanf("%d",&n);

for(i=0;i<n;i++)
{
printf("enter data list : ");
scanf("%d",&dataIn);
addNode(pList3,dataIn);
}

printf("enter the number of terms to put in list 1: ");
scanf("%d",&m);

split(m,n);
printList(pList1);
printList(pList2);
}
```