**CODE:**

#include<stdio.h>

#include<malloc.h>

typedef struct node

{

int data;

struct node \*next;

}NODE;

typedef NODE\* NODEPTR;

NODEPTR start,newNode,tptr,shadow,tptr2,newNode2,start2;

int count=0;

void insert(int givenData)

{

newNode=(NODEPTR)malloc(sizeof(NODEPTR\*));

newNode->data=givenData;

newNode->next=NULL;

if(start==NULL)

{

start=newNode;

}

else

{

for(tptr=start;tptr&&tptr->data<givenData;shadow=tptr,tptr=tptr->next);

if(tptr==start)

{

newNode->next=tptr;

start=newNode;

}

else if(tptr==NULL)

{

shadow->next=newNode;

}

else

{

shadow->next=newNode;

newNode->next=tptr;

}

}

}

void insert2(int givenData)

{

newNode2=(NODEPTR)malloc(sizeof(NODEPTR\*));

newNode2->data=givenData;

newNode2->next=NULL;

if(start2==NULL)

{

start2=newNode2;

}

else

{

for(tptr=start2;tptr&&tptr->data<givenData;shadow=tptr,tptr=tptr->next);

if(tptr==start2)

{

newNode2->next=tptr;

start2=newNode2;

}

else if(tptr==NULL)

{

shadow->next=newNode2;

}

else

{

shadow->next=newNode2;

newNode2->next=tptr;

}

}

}

void display(NODEPTR start)

{

printf("\n");

for(tptr=start;tptr!=NULL;tptr=tptr->next)

{

printf("%d\t",tptr->data);

}

printf("\n");

}

void merge(int a,int b)

{

int i;

tptr=start;

tptr2=start;

for(i=1;i<(a-1);i++)

{

tptr=tptr->next;

}

for(i=1;i<=b;i++)

{

tptr2=tptr2->next;

}

tptr->next=start2;

for(tptr=start,shadow=NULL;tptr;shadow=tptr,tptr=tptr->next);

shadow->next=tptr2;

printf("Merged List\n");

display(start);

}

int main()

{

int num1,num2,a,b;

do

{

scanf("%d",&num1);

if(num1==-1)

{

break;

}

else

{

count++;

insert(num1);

}

}while(1);

display(start);

printf("\n");

do

{

scanf("%d",&num2);

if(num2==-1)

{

break;

}

else

{

insert2(num2);

}

}while(1);

display(start2);

printf("Enter the start node and deletion node for merge\n");

scanf("%d %d",&a,&b);

if(a<count && b<count && a!=b)

{

merge(a,b);

}

else

{

printf("Out of Bounce exception");

}

}

**OUTPUT:**

