10.Write a C program to implement Linked list operations.

Program:

#include<stdio.h>

#include<stdlib.h>

struct Node;

typedef struct Node \* PtrToNode;

typedef PtrToNode List;

typedef PtrToNode Position;

struct Node

{

int e;

Position next;

};

void Insert(int x, List l, Position p)

{

Position TmpCell;

TmpCell = (struct Node\*) malloc(sizeof(struct Node));

if(TmpCell == NULL)

printf("Memory out of space\n");

else

{

TmpCell->e = x;

TmpCell->next = p->next;

p->next = TmpCell;

}

}

int isLast(Position p)

{

return (p->next == NULL);

}

Position FindPrevious(int x, List l)

{

Position p = l;

while(p->next != NULL && p->next->e != x)

p = p->next;

return p;

}

void Delete(int x, List l)

{

Position p, TmpCell;

p = FindPrevious(x, l);

if(!isLast(p))

{

TmpCell = p->next;

p->next = TmpCell->next;

free(TmpCell);

}

else

printf("Element does not exist!!!\n");

}

void Display(List l)

{

printf("The list element are :: ");

Position p = l->next;

while(p != NULL)

{

printf("%d -> ", p->e);

p = p->next;

}

}

void Merge(List l, List l1)

{

int i, n, x, j;

Position p;

printf("Enter the number of elements to be merged :: ");

scanf("%d",&n);

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

{

p = l1;

scanf("%d", &x);

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

p = p->next;

Insert(x, l1, p);

}

printf("The new List :: ");

Display(l1);

printf("The merged List ::");

p = l;

while(p->next != NULL)

{

p = p->next;

}

p->next = l1->next;

Display(l);

}

int main()

{

int x, pos, ch, i;

List l, l1;

l = (struct Node \*) malloc(sizeof(struct Node));

l->next = NULL;

List p = l;

printf("LINKED LIST IMPLEMENTATION OF LIST ADT\n\n");

do

{

printf("\n\n1. INSERT\t 2. DELETE\t 3. MERGE\t 4. PRINT\t 5. QUIT\n\nEnter the choice :: ");

scanf("%d", &ch);

switch(ch)

{

case 1:

p = l;

printf("Enter the element to be inserted :: ");

scanf("%d",&x);

printf("Enter the position of the element :: ");

scanf("%d",&pos);

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

{

p = p->next;

}

Insert(x,l,p);

break;

case 2:

p = l;

printf("Enter the element to be deleted :: ");

scanf("%d",&x);

Delete(x,p);

break;

case 3:

l1 = (struct Node \*) malloc(sizeof(struct Node));

l1->next = NULL;

Merge(l, l1);

break;

case 4:

Display(l);

break;

}

}

while(ch<5);

return 0;

}

Output: 