2.

#include <stdio.h>

#include <stdlib.h>

struct node{

struct node\*next;

int data;

};

struct node \* Union(struct node \* L1, struct node \* L2){

struct node \* output = NULL;

struct node \* outTail = NULL;

while(L1&&L2){

struct node \* newNode = (struct node \*) malloc(sizeof(struct node));

newNode->next = NULL;

if(L1->data<L2->data){

newNode->data = L1->data;

L1 = L1->next;

}

else if(L1->data>L2->data){

newNode->data = L2->data;

L2 = L2->next;

}

else{

int data = L1->data;

newNode->data = data;

while(L1 && L2 && L1->data == data && L2->data == data){

L1 = L1->next;

L2 = L2->next;

}

}

if(!output)

output = outTail = newNode;

else{

outTail->next = newNode;

outTail = outTail->next;

}

}

while(L1){

outTail->next = (struct node \*) malloc(sizeof(struct node));

outTail = outTail->next;

outTail->data = L1->data;

L1 = L1->next;

}

while(L2){

outTail->next = (struct node \*) malloc(sizeof(struct node));

outTail = outTail->next;

outTail->data = L2->data;

L2 = L2->next;

}

outTail->next = NULL;

return output;

}

struct node \* intersection(struct node \* L1, struct node\* L2){

if(L1 == NULL || L2 == NULL)

return NULL;

struct node \* output = NULL;

struct node \* outTail = NULL;

while(L1&&L2){

if(L1->data<L2->data){

L1 = L1->next;

}

else if(L2->data<L1->data){

L2 = L2->next;

}

else{

int data = L1->data;

struct node \* newNode = (struct node \*) malloc(sizeof(struct node));

newNode->data = data;

newNode->next = NULL;

if(output == NULL){

outTail = output = newNode;

}

else{

outTail->next = newNode;

outTail = outTail->next;

}

while(L1 && L2 && L1->data == data && L2->data == data){

L1 = L1->next;

L2 = L2->next;

}

}

}

return output;

}

struct node \* createList(int listNum){

struct node \* list = NULL;

struct node \* list\_tail = NULL;

printf("Enter elements of List %d in increasing order\n",listNum);

char ch = 'y';

do{

int data;

printf("Enter element : ");

scanf("%d",&data);

struct node \* newNode = (struct node \*) malloc(sizeof(struct node));

newNode->data = data;

newNode->next = NULL;

if(list == NULL){

list = list\_tail = newNode;

}

else{

list\_tail->next = newNode;

list\_tail = list\_tail->next;

}

printf("Would you like to insert another element [Y/N] : ");

scanf(" %c",&ch);

}while(ch == 'y' || ch == 'Y');

return list;

}

void print(struct node \* list){

if(list == NULL){

printf("Empty List\n");

return;

}

while(list!=NULL){

printf("%d ",list->data);

list = list->next;

}

printf("\n");

}

int main() {

struct node \* L1 = NULL;

struct node \* L2 = NULL;

struct node \* L3 = NULL;

struct node \* L4 = NULL;

L1 = createList(1);

L2 = createList(2);

printf("List 1 : ");

print(L1);

printf("List 2 : ");

print(L2);

printf("Union : ");

L3 = Union(L1, L2);

print(L3);

printf("Intersection : ");

L4 = intersection(L1, L2);

print(L4);

return 0;

}

Output

List 1 : 2 3

List 2 : 4

Would you like to insert another el

ement [Y/N] : Y

Enter element: 1

Enter element : 2

Would you like to insert another el

ement [Y/N] : N

Enter elements of List 2 in increas

ing order

Enter element : 3

Would you like to insert another el

ement [Y/N] : N

List 1 : 1 2

List 2 : 3

Union : 1 2 3

Intersection : Empty List

Union : 2 3 4

Intersection : Empty List