//Long Integer addition using Circular Doubly Linked List

//with header node

#include <stdio.h>

#include<stdlib.h>

typedef struct node \*Nodeptr;

struct node{

int data;

Nodeptr llink, rlink;

};

Nodeptr getnode(){

Nodeptr temp;

temp= (Nodeptr) malloc(sizeof(struct node));

return temp;

}

void InsertFront(int item, Nodeptr head){

Nodeptr temp,next;

temp = getnode();

temp->data = item;

next = head->rlink;

temp->llink = head;

head->rlink = temp;

temp->rlink = next;

next->llink = temp;

}

void Display(Nodeptr head){

Nodeptr temp;

for(temp=head->rlink; temp!=head;temp=temp->rlink)

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

}

//Function creates linked list representing the long integer

Nodeptr ReadLongInteger(){

Nodeptr head;

char str[100];

int i,n;

printf("Enter the string representing long integer : ");

scanf("%s", str);

for(n=0;str[n];n++);

head = getnode();

head->llink = head->rlink = head;

//Extract each digit from left and insert at the front of the list

for(i=n-1;i>=0;i--)

InsertFront(str[i]-'0', head);

return head;

}

Nodeptr AddLongInteger(Nodeptr A,Nodeptr B){

int digit,sum,carry;

Nodeptr head,r,R,a,b;

carry = 0;

a=A->llink;

b=B->llink;

head = getnode();

head->llink= head->rlink = head;

while(a!= A && b != B){

sum = a->data + b->data + carry;

digit = sum%10;

carry = sum/10;

InsertFront(digit,head);

a = a->llink;

b = b->llink;

}

//Identify the bigger number

if (a!= A){

r=a; R=A;

}

else {

r=b; R=B;

}

//add carry to remaining digits of bigger number

while(r!= R){

sum = r->data + carry;

digit = sum%10;

carry = sum/10;

InsertFront(digit,head);

r = r->llink;

}

//Insert the last carry, if present.

if (carry)

InsertFront(carry,head);

return head;

}

int main(){

Nodeptr a,b,sum;

a = ReadLongInteger();

b = ReadLongInteger();

printf("\nhead Integer : \n");

Display(a);

printf("\nSecond Integer : \n");

Display(b);

sum=AddLongInteger(a,b);

printf("\nSum of 2 given Integers : \n");

Display(sum);

return 0;

}