

#include<iostream>

using namespace std;

class Node{

public:

int data;

Node\*next;

Node\*prev;

}

typedef Node node;

void push(Node\*\*head\_ref,int new\_data);

void printList (Node \*n);

void swapPointer(Node \*\*a, Node\*\*b);

int getSize(Node\* node);

node\*addSameSize(Node \* head1,Node\*head2,int \*carry);

void addCarryToRemaining(Node\*head1, Node\*cur,int \* carry,Node\*\* result);

void addList(Node\* head1,Node\*head2,Node\*\*result)

int main()

{

Node \* head1=NULL,\*head2 = NULL,\*result=NULL;

int arr1[]={9,9,9};

int arr2[]={1,8};

int size1=sizeof(arr1)/sizeof(arr1[0]);

int size2=sizeof(arr2)/sizeof(arr2[0]);

int i;

for(i=size1-1;i>=0;i--)

push(&head1,arr1[i]);

for(i=size2-1;i>=0;i--)

push(&head2,arr2[i]);

addList(head1,head2,&result);

printList(result);

}

node\*addSameSize(Node\* head1,Node\*head2,int \* carry)

{

if(head1==NULL) return NULL;

int sum;

Node \* Result = new Node[(sizeof(Node))];

result->next = addSameSize(head1>next,head2->next,carry)

sum=head1->data + head2->data + \*carry;

\*cfarry=sum/10;

sum=sum%10;

result->data = sum;//a

return result;

}

void addCarryToRemaining(Node\* head1, Node \*cur,int \* carry,Node\*\* result)

{

int sum;

if(head1!=cur)

{addCarryToRemaining(head1->next, cur,carry,result);

sum=head1->data + \*carry;

\*carry=sum/10;

sum=sum%10;

push(result,sum);

}

}

void addList(Node\* head1,Node\*head2,Node\*\*result)

{

Node\*cur;

if(head1==NULL)

{

\*result=head2; return;

}

else if(head2==NULL)

{

\*result=head1; return;

}

int size1=getSize(head1);

int size2=getSize(head2);

int carry=0;

if(sixe1==size2)

\*result = addSaneSize(head1,head2,&carry);

else{

int diff = abs(size1 - size2);

if(size1<size2)

swapPointer(&head1,&head2);

for(cur=head1;dif--;cur=cur->next);

\*result=addSameSize(cur,,head2,&carry);

addCarryToRemaining(head1,cur,&carry,result);

}

if(carry)

push(result,carry);

}

int getSize(Node\*node)

{

int size==0;

while(node!=NULL)

{

node=node->nexr;

size++;

}

return size;

}

void swapPointer(Node \*\*a,Node \*\*b)

{

Node\*t=\*a;

\*a=\*b;

\*b=t;

}

void printList(Node \*node)

{

while(node!=NULL)

{

cout<<node->data<<" ";

node=node->next;

}

cout<<endl;

}

void push(Node\*\*head\_ref,int new\_data)

{

Node \* new\_node=new Node[(sizeof(Node))];

new\_node->data=new\_data;

new\_node->next=(\*head\_ref);

(\*head\_ref)=new\_node;

}















