#include<iostream>

using namespace std;

struct Node

{

int bin;

Node \*next,\*prev;

Node(){}

Node(int n)

{

bin=n;

next=NULL;

prev=NULL;

}

};

Node \*createnode(){

Node \*ntemp=new Node;

cout<<"Enter Binary :- ";

cin>>ntemp->bin;

ntemp->next=NULL;

ntemp->prev=NULL;

return ntemp;

}

class Binary

{

public:

Node \*start;

Binary()

{

start=NULL;

}

void create();

void display();

void compliment();

void sum(Binary);

};

void Binary::create()

{

int n;

cout<<"Eneter the No. of Bits :- ";

cin>>n;

while(start!=NULL)

{

Node \*old=start;

start=start->next;

delete old;

}

for(int i=0;i<n;i++)

{

Node \*ntemp=createnode();

if(start==NULL)

start=ntemp;

else{

Node \*trav=start;

while(trav->next!=NULL)

trav=trav->next;

ntemp->prev=trav;

trav->next=ntemp;

}

}

cout<<"\nBinarry Inserted :- "<<endl;

display();

}

void Binary::display()

{

Node \*trav=start;

while(trav!=NULL)

{

cout<<trav->bin<<" ";

trav=trav->next;

}

cout<<endl;

}

void Binary::compliment()

{

Binary ones,twos;

Node \*trav=start;

//Code For 1's Compliment

while(trav!=NULL)

{

Node \*ntemp=new Node(!trav->bin);

if(ones.start==NULL)

ones.start=ntemp;

else{

Node \*trav1=ones.start;

while(trav1->next!=NULL){

trav1=trav1->next;

}

ntemp->prev=trav1;

trav1->next=ntemp;

}

trav=trav->next;

}

//Code For 2's Compliment

trav=ones.start;

while(trav->next!=NULL)

trav=trav->next;

int carry=1;

while(trav!=NULL)

{

int sum=((trav->bin==0 && carry==0) || (trav->bin==1 && carry ==1)) ? 0 : 1;

carry=(trav->bin==1 && carry==1) ? 1 : 0;

Node \*ntemp=new Node(sum);

ntemp->next=twos.start;

twos.start=ntemp;

trav=trav->prev;

}

cout<<"\n1's Compliment :- ";

ones.display();

cout<<"\n2's Compliment :- ";

twos.display();

}

void Binary::sum(Binary bin2)

{

if(start==NULL)

{

cout<<"Binary 1 is Empty "<<endl;

return ;

}

else if(bin2.start==NULL)

{

cout<<"Binary 2 is Empty "<<endl;

return ;

}

else{

Binary ans;

int b1,b2,carry=0;

Node \*trav1=start;

Node \*trav2=bin2.start;

while(trav1->next!=NULL)

trav1=trav1->next;

while(trav2->next!=NULL)

trav2=trav2->next;

while(trav1!=NULL || trav2!=NULL)

{

if(trav1!=NULL)

{

b1=trav1->bin;

trav1=trav1->prev;

} else{ b1=0; }

if(trav2!=NULL)

{

b2=trav2->bin;

trav2=trav2->prev;

} else{ b2=0; }

int sum=b1^b2^carry;

carry=((b1 && b2 && carry) || (!b1 && b2 && carry) || (b1 && !b2 && carry) || (b1 && b2 && !carry));

Node \*ntemp=new Node(sum);

ntemp->next=ans.start;

ans.start=ntemp;

}

if(carry==1){

Node \*ntemp=new Node(carry);

ntemp->next=ans.start;

ans.start=ntemp;

}

cout<<"\nSUM :- ";

ans.display();

}

}

//============Main Function============//

int main()

{

Binary n1,n2;

int choice;

do

{

cout<<"\n1.Input Binary 1 \n2.Input Binary 2 \n3.1's Compliment and 2's Compliment of Binary 1 \n4.1's Compliment and 2's Compliment of Binary 2 \n5.Binary Sum of both \n0.Exit \nEnter Choice :- ";

cin>>choice;

switch(choice)

{

case 1:

n1.create();

break;

case 2:

n2.create();

break;

case 3:

n1.compliment();

break;

case 4:

n2.compliment();

break;

case 5:

n1.sum(n2);

break;

}

} while (choice!=0);

}