```
//DSL Lab 07 Binary Linked List



#include<iostream>
#include<string>
using namespace std;

struct node{
int data;
struct node *next;
struct node *prev;
};

class doublyLikedList{

public:
node *header , *footer;
int bits;

doublyLikedList(){
header = NULL;
footer = NULL;
}

node * input(){
header = NULL;
node *x, *y;

cout<<"\nEnter number of bits in binary numbers: ";
cin>>bits;
int var;
for(int i=0;i<bits;i++){
    x = new node;
    cin>>var;
    if(var<=0){
        x->data = 0;
    }else{
    x->data = 1;
    }
    x->next = NULL;
    x->prev = NULL;

    if(header==NULL){
        header=x;
        footer=x;
        }
    else{
        y=header;
        while(y->next!=NULL){
            y=y->next;
        }

        y->next=x;
               x->prev=y;
        footer=x;
        }
    }
    return header;
}

void display(node *d){
    cout<<"\n\nEntered binary number is: "<<endl;
    node *y;
    y = d;
    while(y!=NULL){
        cout<<y->data;
        y=y->next;
    }
}

void onesCompli(){
    cout<<"\n\nOne's Complement is: "<<endl;
        node *y;
        y=header;
               while(y!=NULL){
                       if(y->data==0){
                cout<<1;
            }
                       else{
            cout<<0;
                       }
                       y=y->next;
                       }
}

void inverse(node *firstNonZero, string answer, int bits){
node *y;

y = firstNonZero;

while(y!=NULL){
    if(y->data == 0){
        //cout<<1;
        answer += '1';
    }else{
    //cout<<0;
       answer += '0';
    }
    y = y->prev;
}
//cout<<answer<<endl;
for (int i=answer.length()-1; i>=0; i--){
cout << answer[i];
}
}
void twosCompli(){
    int arr[bits];
    node *firstNonZero;
cout<<"\n\nTwo's Complement is: "<<endl;
node *y;
string answer;
```

```
y = footer;
while(y!=NULL){

    if(y->data == 1){
        answer += '1';
        firstNonZero = y->prev;
        inverse(firstNonZero, answer, bits);
        break;
    }else{
       answer += '0';
    }
    y = y->prev;

}
}

int BinaryToDecimal(node *bin){
    int twoPowers[11] = {1,2,4,8,16,32,64,128,256,512,1024};
    node *xfirst;
    xfirst = bin;
    int i = 0;
    int ans = 0;
    while(xfirst != NULL){
        ans += xfirst->data * twoPowers[i];
        i++;
        xfirst = xfirst->prev;
    }
    return ans;
}
void DecimalToBinary(int n){
    int binaryNum[32];
    int i = 0;
    while (n > 0) {
        binaryNum[i] = n % 2;
        n = n / 2;
        i++;
    }
    for (int j = i - 1; j >= 0; j--){
        cout << binaryNum[j];
    }
}
void add(node *bin1, node *bin2){

        node *binary1,*binary2;
        binary1 = bin1;
        binary2 = bin2;
        cout<<"\nAddition is\n";
        int sum = BinaryToDecimal(binary1) + BinaryToDecimal(binary2);
        DecimalToBinary(sum);
    }

};
int main(){
    doublyLikedList obj;
node *z, *zf;

doublyLikedList obj1;
```

```
node *z1, *z1f;
    cout<<"<<<<<<<<<< Enter first binary number >>>>>>>>>>>"<<endl;
obj.input();

    cout<<"\n<<<<<<<<<<< Enter second binary number >>>>>>>>>>"<<endl;
    obj1.input();

z = obj.header;
obj.display(z);
obj.onesCompli();
obj.twosCompli();
zf = obj.footer;


z1 = obj1.header;
obj1.display(z1);
obj1.onesCompli();
obj1.twosCompli();
z1f = obj1.footer;

doublyLikedList addObj;
obj.display(z);
obj1.display(z1);
addObj.add(zf,z1f);
}
```