#include<iostream>

using namespace std;

struct node

{

      int data;

      struct node \*next, \*prev;

};

class cBinary

{

      public:

       struct node \*head;

      cBinary()

      {

           head=NULL;

      }

      struct node \* create(struct node \*);

      void display(struct node \*);

      void ocomp(struct node \*h);

      void oocomp(struct node \*h);

      void add(struct node \*,struct node \*);

};

struct node\* cBinary::create(struct node \*h)

{

      int n,rem;

      struct node \*nn;

      cout<<"\n Enter number : ";

      cin>>n;

      while(n!=0)

      {

            rem=n%2;

            n=n/2;

            nn=new node;

            nn->prev=NULL;

            nn->next=NULL;

            nn->data=rem;

            if(h==NULL)

            {

                  h=nn;

            }

            else

            {

                  h->prev=nn;

                  nn->next=h;

                  h=nn;

            }

      }

      return h;

}

void cBinary::display(struct node \*temp)

{

      while(temp!=NULL)

      {

            cout<<temp->data;

            temp=temp->next;

      }

}

void cBinary::ocomp(struct node \*temp)

{

      struct node \*t=temp;

      while(temp!=NULL)

      {

           if(temp->data==1)

           {

                temp->data=0;

           }

           else

           {

                temp->data=1;

           }

           temp=temp->next;

      }

      cout<<"your one's complemnt is \n";

      display(t);

}

void cBinary::oocomp(struct node \*temp)

{

  struct node \*t=temp;

  int carry=1;

    while(temp->next!=NULL)

    {

        temp=temp->next;

    }

  while(temp!=NULL)

  {

    if(temp->data==1 && carry==1)

    {

     temp->data=0;

     carry=1;

    }

    else if(temp->data==0 && carry==1 )

    {

     temp->data=1;

     carry=0;

    }

    else

    if(carry==0)

    {

     break;

    }

    temp=temp->prev;

  }

     cout<<"your two's complement is";

   display(t);

}

void cBinary::add(struct node \*temp1, struct node \*temp2)

{

    struct node \*temp3=NULL,\*nn;

    int carry=0,res;

    while(temp1->next!=NULL)

    {

        temp1=temp1->next;

    }

    while(temp2->next!=NULL)

    {

        temp2=temp2->next;

    }

    while(temp1!=NULL && temp2!=NULL)

    {

           if(temp1->data==1 && temp2->data==1 && carry==0)

        {

          res=0;

          carry=1;

        }

       else if(temp1->data==1 && temp2->data==1 && carry==1)

       {

      res=1;

      carry=1;

       }

      else if(temp1->data==1 && temp2->data==0 && carry==1)

       {

      res=0;

      carry=1;

       }

       else if(temp1->data==0 && temp2->data==0 && carry==0)

       {

      res=0;

      carry=0;

       }

      else if(temp1->data==0 && temp2->data==1 && carry==1)

       {

      res=0;

      carry=1;

       }

       else if(temp1->data==1 && temp2->data==0 && carry==0)

       {

      res=1;

      carry=0;

       }

      else if(temp1->data==0 && temp2->data==0 && carry==1)

       {

      res=1;

      carry=0;

       }

       else if(temp1->data==0 && temp2->data==1 && carry==0)

       {

      res=1;

      carry=0;

      }

      //\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

      nn=new node;

      nn->data=res;

      nn->prev=NULL;

      if(temp3==NULL)

        temp3=nn;

      else

      {

        temp3->prev=nn;

        nn->next=temp3;

        temp3=nn;

      }

    //\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

    temp2=temp2->prev;

    temp1=temp1->prev;

    }

    if(carry==1)

    {

        nn=new node;

        nn->data=1;

        temp3->prev=nn;

        nn->next=temp3;

        temp3=nn;

        nn->prev=NULL;

    }

    display(temp3);

}

int main()

{

       cBinary b,n1,n2;

       int ch;

  int i;

       cout<<"\n1.CREATE \n2.DISPLAY \n3.ONE'S COMP \n4.Two's complement \n5.Addition of binary number\n6. Exit\n ";

       while(ch!=6)

       {

           cout<<"\n\nEnter the choice : ";

           cin>>ch;

           switch(ch)

           {

                 case 1:

                 b.head=b.create(b.head);

                 break;

                 case 2:

                 b.display(b.head);

                 break;

                 case 3:

                 b.ocomp(b.head);

                 break;

             case 4:

          b.oocomp(b.head);

          break;

                case 6:

                 cout<<"EXIT";

                 break;

            case 5:

                  n1.head=NULL;

                  n2.head=NULL;

                  n1.head=n1.create(n1.head);

                  n2.head=n2.create(n2.head);

              n1.add(n1.head,n2.head);

              break;

                 default:

                 cout<<"\n!!!!!!!Wrong Choice!!!!!";

                 break;

           }

       }

}