# Code:

import java.io.\*;

import java.util.\*;

import java.lang.\*;

/\*\*

\* @author "Tushar Ambelkar"

\* @version 1.0

\*/

public class Comp{

/\*\*

\* This function is used to add two binary numbers in two's complement form.

\* <p>

\* First it converts both input numbers to equal size.

\* Then It adds them bit by bit starting from rightmost bit...

\* <p>

\* An overflow condition is indicated if two numbers are added,

\* and they are both positive or both negative,

\* then overflow occurs if and only if the result has the opposite sign.

\*

\* @param in1 First input binary number in two's complement form

\* @param in2 Second input binary number in two's complement form

\* @return Sum of input in two's complement form.

\*/

public static String add(String in1,String in2){

String output;

StringBuilder sb=new StringBuilder();

int a,b;

a=in1.length();

b=in2.length();

if(a<b){

for(int i=0;i<b-a;i++)

sb.append(in1.charAt(0));

sb.append(in1);

in1=sb.toString();

}

else if(a>b){

for(int i=0;i<a-b;i++)

sb.append(in2.charAt(0));

sb.append(in2);

in2=sb.toString();

}

sb=new StringBuilder();

int c=0;

int s=0;

for(int i=in1.length()-1;i>=0;i--){

a=Integer.parseInt(new Character(in1.charAt(i)).toString());

b=Integer.parseInt(new Character(in2.charAt(i)).toString());

s=a^b^c;

c=(a&b)|(c&(a^b));

sb.append(s);

}

sb=sb.reverse();

boolean n=false;

if(in1.charAt(0)==in2.charAt(0)&&sb.charAt(0)!=in2.charAt(0)){

System.out.println("Overflow....");

n=true;

sb=sb.reverse();

sb.append(in1.charAt(0));

sb=sb.reverse();

}

if(n&&c==1){

sb=sb.reverse();

sb.append(c);

sb=sb.reverse();

}

output=sb.toString();

return output;

}

/\*\*

\* This function calculates the two's complement of input inString and returns it.

\*

\* @param inString input binary string....

\* @return twos complement of input.

\*/

public static String twoComplement(String inString){

String output;

StringBuilder sb=new StringBuilder();

for(int i=0;i<inString.length();i++){

if(inString.charAt(i)=='0')

sb.append('1');

else

sb.append('0');

}

//System.out.print(sb.toString());

output = add(sb.toString(),"01");

//System.out.print(output);

return output;

}

/\*\*

\* This is the main function which starts the execution of program.....

\*/

public static void main(String []args) {

Scanner sc =new Scanner(System.in);

boolean k=true;

int choice;

while(k){

System.out.println("What would you like to do:\n1:Addition\n2:Subtraction\n3:Exit\nEnter appropriate choice....\n");

choice=sc.nextInt();

String a,b,out;

switch (choice){

case 1:

System.out.println("Enter values in twos complement form of representation only");

System.out.println("Enter first binary input:");

a =sc.next(); //Input first binary number...

if(!(a.matches("[10]+"))){

System.out.println("Wrong input format...");break; //Check for validity of binary input

}

System.out.println("Enter second binary input:");

b=sc.next(); //Input second binary number...

if(!(b.matches("[10]+"))){

System.out.println("Wrong input format...");break; //Check for validity of binary input

}

out=add(a,b); //call Comp.add() static function...

System.out.println("The result of operation is:" +out);

break;

case 2:

System.out.println("Enter values in twos complement form of representation only");

System.out.println("Enter first binary input:");

a =sc.next(); //Input first binary number...

if(!(a.matches("[10]+"))){

System.out.println("Wrong input format...");break; //Check validity of binary input

}

System.out.println("Enter second binary input:");

b=sc.next(); //Input second binary number...

if(!(b.matches("[10]+"))){

System.out.println("Wrong input format...");break; //Check validity of binary input

}

//First evaluate two's complement by calling Comp.twoComplement()...

//then call Comp.add() to add subtraend with 2 complement of minuend.

out=add(a,twoComplement(b));

System.out.println("The result of operation is:" +out);

break;

case 3:

System.out.println("Terminating the program.......\nBye");

k=false;

break;

default :

System.out.println("Wrong input entered.....\n");

break;

}

}

}

}

# Output:

