import java.io.\*;

class Crc

{

public static void main(String args[]) throws IOException

{

BufferedReader br=new BufferedReader(new InputStreamReader(System.in));

int[ ] data;

int[ ]div;

int[ ]divisor;

int[ ]rem;

int[ ]crc;

int data\_bits, divisor\_bits, tot\_length;

System.out.println("Enter number of data bits : ");

data\_bits=Integer.parseInt(br.readLine());

data=new int[data\_bits];

System.out.println("Enter data bits : ");

for(int i=0; i<data\_bits; i++)

data[i]=Integer.parseInt(br.readLine());

System.out.println("Enter number of bits in divisor : ");

divisor\_bits=Integer.parseInt(br.readLine());

divisor=new int[divisor\_bits];

System.out.println("Enter Divisor bits : ");

for(int i=0; i<divisor\_bits; i++)

divisor[i]=Integer.parseInt(br.readLine());

tot\_length=data\_bits+divisor\_bits-1;

div=new int[tot\_length];

rem=new int[tot\_length];

crc=new int[tot\_length];

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

div[i]=data[i];

System.out.print("Dividend (after appending 0's) are : ");

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

System.out.print(div[i]);

*System.out.println();*

for(int j=0; j<div.length; j++)

{

rem[j] = div[j];

}

rem=divide(div, divisor, rem);

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

{

crc[i]=(div[i]^rem[i]);

}

*System.out.println();*

System.out.println("CRC code : ");

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

System.out.print(crc[i]);

*System.out.println();*

System.out.println("Enter CRC code of "+tot\_length+" bits : ");

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

crc[i]=Integer.parseInt(br.readLine());

for(int j=0; j<crc.length; j++)

{

rem[j] = crc[j];

}

rem=divide(crc, divisor, rem);

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

{

if(rem[i]!=0)

{

System.out.println("Error");

break;

}

if(i==rem.length-1)

System.out.println("No Error");

}

}

static int[] divide(int div[],int divisor[], int rem[])

{

int cur=0;

while(true)

{

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

rem[cur+i]=(rem[cur+i]^divisor[i]);

while(rem[cur]==0 && cur!=rem.length-1)

cur++;

if((rem.length-cur)<divisor.length)

break;

}

return rem;

} }

**Output:**

