

## Java practicals

### Question 2

#### 2a

```
import java.util.*;

public class Main
{
    public static void main(String[] args) {
        Scanner sc= new Scanner(System.in);
        System.out.print("Enter 1st binary number:");
        String b1= sc.nextLine();
        System.out.print("Enter 2nd binary number:");
        String b2= sc.nextLine();
        String b3="";//result
        int m,n,xoresult,len,carry=0,temp,len_diff;
        char ch,ch2,ch3;
        //equate lengths
        if(b1.length()>b2.length()){
            len_diff=b1.length()-b2.length();

            for(int i=0;i<len_diff;i++){
                //b5+=b4.concat("0");
                b2=0+b2;
            }
            //b2=b5+b2;
            len=b1.length();
        }
        else{
            len_diff=b2.length()-b1.length();
            for(int i=0;i<len_diff;i++){
```

```
        b1=0+b1;
    }
```

```
        len=b2.length();
    }
```

```
//binary addition
```

```
for(int i=len-1;i>=0;i--){ //start from unit place,hence from behind
```

```
    ch=b1.charAt(i);
```

```
    m=ch-'0';
```

```
    ch2=b2.charAt(i);
```

```
    n=ch2-'0';
```

```
    if(carry==1){//if carry is generated in between the numbers ie past xor
```

```
        temp=carry^m;
```

```
        //to check whether the carry is generated by xoring carry and m
```

```
        //if carry is generated here then carry is not generated when xoring temp and n
```

```
        if(carry==1 & m==1){
```

```
            carry=1;
```

```
        }
```

```
        else {
```

```
            carry=0;
```

```
        }
```

```
        xresult=temp^n;
```

```
        // 1 (carry) + 1(m) + 1(n) --> this is broken down into 1+1=0(temp) carry 1, then
```

```
0(temp)+1(n),
```

```

//either carry+m might generate carry or temp+n, but not both
//then check if temp+n generates carry if carry+m doesnt generate carry
if(temp==1 & n==1 ){
    carry=1;
}

```

```

b3+=xoresult;

```

```

//if bit is msb
if(carry==1 & i==0){
    b3+=carry;
}

}

```

```

else{
    //this condition is when no carry is arrives from past xor operation
    xoresult=m^n;
    //to check whether the carry is generated by this xor operation
    if(m==1&n==1){
        carry=1;
    }
    else {
        carry=0;
    }
    b3+=xoresult;

```

```

//if bit is msb
if(i==0 & carry==1){

```

```

        b3+=carry;
    }

}

}

System.out.println("the addition is:");
//the resultant string is stored reverse, hence reverse string to get proper output.
for(int i=b3.length()-1;i>=0;i--){

    System.out.print(b3.charAt(i));

}

}

}

```

## 2b

```

import java.util.*;

public class Main
{

    public static void main(String[] args){

        Scanner sc=new Scanner(System.in);

        System.out.println("enter 1 for dec_to_binary or 2 for binary_to_dec");
        int c=sc.nextInt();

        if(c==1){

            System.out.println("enter a decimal number:");
            int a =sc.nextInt();

```

```
int i,k=0;

int binary[]=new int[10];

while(a>1){

i=a%2;

binary[k]=i;

k++;

a=a/2;

}

binary[k]=a;

System.out.print("binary is:");

for(i=k;i>=0;i--){

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

}

System.out.println();

}

else{

//binary to decimal

System.out.println("enter a binary number:");

int bin =sc.nextInt();

int i=0,k=0;//k and i are already defined

int deci_sum=0,power;

while(bin!=0){

    k=bin%10;

    power=(int)Math.pow(2,i);

    k=k*power;

    deci_sum+=k;

    bin=bin/10;

}
```

```

        i++;

    }

    System.out.println("decimal number is:"+dec_i_sum);
}

}
}

```

## 2c

```

import java.util.*;

public class Main
{
    public static void main(String[] args) {
        System.out.println("Enter a string");
        Scanner sc= new Scanner(System.in);
        String in_user=sc.next();
        String reversed ="";
        char b;

        for(int i=in_user.length()-1;i>=0;i--){
            b=in_user.charAt(i);
            reversed+=b;

        }
        System.out.println("the reversed string is:"+reversed);
    }
}

```