

Problem Statement –

3. Joseph is learning digital logic subject which will be for his next semester. He usually tries to solve unit assignment problems before the lecture. Today he got one tricky question. The problem statement is “A positive integer has been given as an input. Convert decimal value to binary representation. Toggle all bits of it after the most significant bit including the most significant bit. Print the positive integer value after toggling all bits”.

Constrains-

$1 \leq N \leq 100$

Example 1:

Input :

10 -> Integer

Output :

5 -> result- Integer

Explanation:

Binary representation of 10 is 1010. After toggling the bits(1010), will get 0101 which represents “5”. Hence output will print “5”.

```
import java.util.*;
```

```

class Main
{
    public static void main(String[] args)
    {
        Scanner sc=new Scanner(System.in);
        int no=sc.nextInt();
        String bin="";

        while(no!=0)
        {
            bin=(no&1)+bin;
            no=no>>1;
        }
        bin=bin.replaceAll("1","2");
        bin=bin.replaceAll("0","1");
        bin=bin.replaceAll("2","0");
        int res=Integer.parseInt(bin,2);
        System.out.println(res);
    }
}

```

4. Jack is always excited about sunday. It is favourite day, when he gets to play all day. And goes to cycling with his friends.

So every time when the months starts he counts the number of sundays he will get to enjoy. Considering the month can start with any day, be it Sunday, Monday.... Or so on.

Count the number of Sunday jack will get within n number of days.

Example 1: Input

mon-> input String denoting the start of the month.

13 -> input integer denoting the number of days from the start of the month.

Output :

2 -> number of days within 13 days.

Explanation:

The month start with mon(Monday). So the upcoming sunday will arrive in next 6 days. And then next Sunday in next 7 days and so on.

Now total number of days are 13. It means 6 days to first sunday and then remaining 7 days will end up in another sunday. Total 2 sundays may fall within 13 days.

```
import java.util.*;
class Main
{
    public static void main(String[] args)
    {
        Scanner sc=new Scanner(System.in);
        String str=sc.next(); //mon
```

```

        int n=sc.nextInt(); //13
String
arr[]={"mon","tue","wed","thu","fri","sat","sun"};
        int i=0;
        for(i=0;i< arr.length;i++)
            if(arr[i].equals(str))
                break;
        int res=1;
        int rem=6-i; //rem = 6
        n=n-rem;    //n= 13-6 = 7
        if(n >0){
            res+=n/7;
            System.out.println(res);    }
    else{ System.out.println("No sunday");}
} }

```

5. Airport security officials have confiscated several item of the passengers at the security check point. All the items have been dumped into a huge box (array). Each item possesses a certain amount of risk[0,1,2]. Here, the risk severity of the items represent an array[] of N number of integer values. The task here is to sort the items based on their levels of risk in the array. The risk values range from 0 to 2.

Example :

Input :

7 -> Value of N

[1,0,2,0,1,0,2]-> Element of arr[0] to arr[N-1], while input each element is separated by new line.

Output :

0 0 0 1 1 2 2 -> Element after sorting based on risk severity

Example 2:

input : 10 -> Value of N

[2,1,0,2,1,0,0,1,2,0] -> Element of arr[0] to arr[N-1], while input each element is separated by a new line.

Output :

0 0 0 0 1 1 1 2 2 2 ->Elements after sorting based on risk severity.

Explanation:

In the above example, the input is an array of size N consisting of only 0's, 1's and 2s. The output is a sorted array from 0 to 2 based on risk severity.

```
import java.util.*;  
class Main  
{  
    public static void main(String[] args)
```

```

{
Scanner sc=new Scanner(System.in);
int n=sc.nextInt();
int arr[]=new int[n];
for(int i=0;i< n;i++)
    arr[i]=sc.nextInt();
int countZero=0,countOne=0,countTwo=0;
for(int i=0;i< n;i++)
{
    if(arr[i]==0)
        countZero++;
    else if(arr[i]==1)
        countOne++;
    else if(arr[i]==2)
        countTwo++;
}
int j =0;
while(countZero >0)
{
    arr[j++]=0;
    countZero--;
}
while(countOne >0)
{
    arr[j++]=1;
    countOne--;
}

while(countTwo >0)

```

```

        {
            arr[j++]=2;
            countTwo--;
        }

    for(int i=0;i < n;i++)
        System.out.print(arr[i]+" ");
    }
}

```

Given an integer array Arr of size N the task is to find the count of elements whose value is greater than all of its prior elements.

Note : 1st element of the array should be considered in the count of the result.

For example,

N=5

Arr[]={7,4,8,2,9}

As 7 is the first element, it will consider in the result.

8 and 9 are also the elements that are greater than all of its previous elements.

Since total of 3 elements is present in the array that meets the condition.

Hence the output = 3.

Example 1:

Input

5 -> Value of N, represents size of Arr

7-> Value of Arr[0]

4 -> Value of Arr[1]

8-> Value of Arr[2]

2-> Value of Arr[3]

9-> Value of Arr[4]

Output :

3

Example 2:

5 -> Value of N, represents size of Arr

3 -> Value of Arr[0]

4 -> Value of Arr[1]

5 -> Value of Arr[2]

8 -> Value of Arr[3]

9 -> Value of Arr[4]

Output :

5

Constraints

$1 \leq N \leq 20$

$1 \leq \text{Arr}[i] \leq 10000$

```
import java.util.*;
class Main
{
    public static void main(String[] args)
    {
        Scanner sc=new Scanner(System.in);
        int n=sc.nextInt();
        int arr[]=new int[n];
        for(int i=0;i< n;i++)
            arr[i]=sc.nextInt();
        int max=Integer.MIN_VALUE;

        int count=0;
        for(int i=0;i< n;i++)
```

```
{
    if(arr[i]>max)

        {
            max=arr[i];
            count++;
        }
}
System.out.println(count);
}
}
```