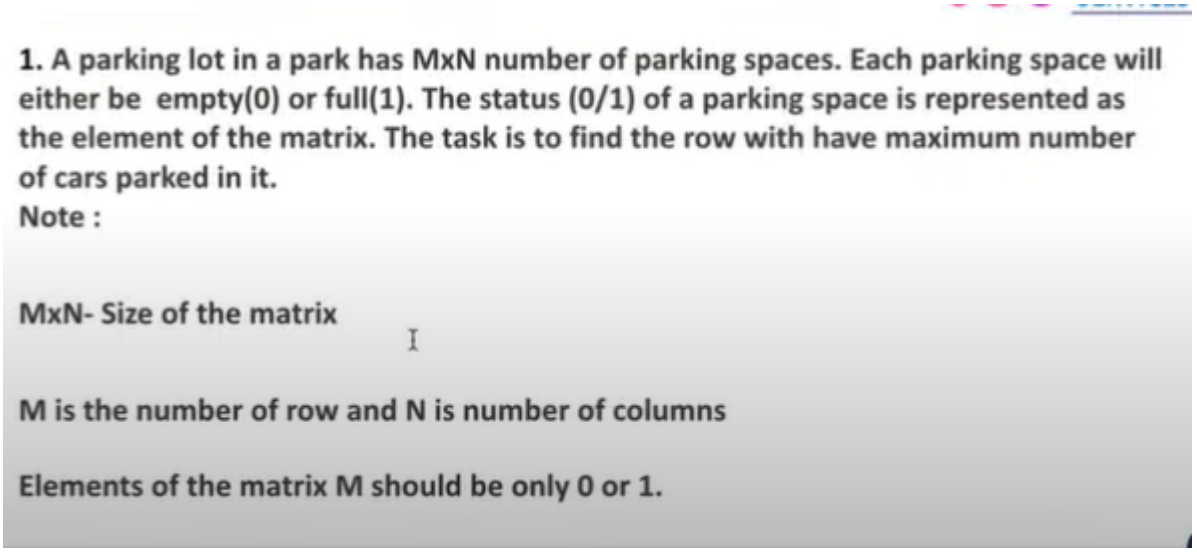
1.



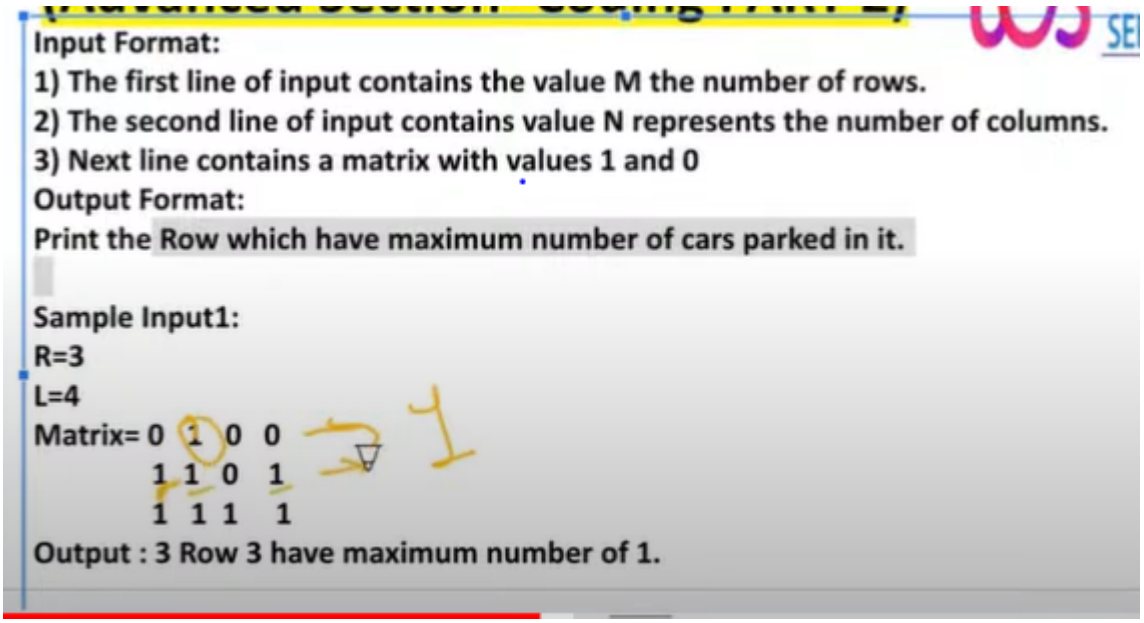
1. A parking lot in a park has MxN number of parking spaces. Each parking space will either be empty(0) or full(1). The status (0/1) of a parking space is represented as the element of the matrix. The task is to find the row with have maximum number of cars parked in it.

Note :

MxN- Size of the matrix

M is the number of row and N is number of columns

Elements of the matrix M should be only 0 or 1.



Input Format:

1) The first line of input contains the value M the number of rows.

2) The second line of input contains value N represents the number of columns.

3) Next line contains a matrix with values 1 and 0

Output Format:

Print the Row which have maximum number of cars parked in it.

Sample Input1:

R=3

L=4

Matrix= 0 1 0 0

1 1 0 1

1 1 1 1

Output : 3 Row 3 have maximum number of 1.

1. ans

```
import java.util.*;
public class Main
{
        public static void main(String[] args) {
        int rowSum =0, target = 0, R=0;
   Scanner sc = new Scanner(System.in);
   System.out.print("M = ");
   int M = sc.nextInt();
   System.out.print("%nN = ");
   int N = sc.nextInt();
        int[][] a = new int[M][N];
   System.out.println("Matrix = ");
   for(int i=0 ; i<M ; i++)
   {
      for(int j=0 ; j<N ; j++)
```

```
            {
                a[i][j] = sc.nextInt();
                rowSum += a[i][j];
            }
            if(rowSum > target)
            {
                target = rowSum;
                R = i+1;
            }
            rowSum = 0;
        }
        System.out.println("Row "+R+" have maximum Number of 1");
            }
    }
```

2.

Given an array consists of n elements you have to print the elements which has appeared even number of times

a) **N size of array**
b) **Next line contains elements of array**
Print the elements which has appeared even number of times

Constraints:

Size of array should be greater than zero
Elements of array should be greater than zero

```
import java.util.*;
class Main{
   public static void main(String[] args)
   {
      int count=0;
      Map<Integer,Integer> Mp = new HashMap<>();
      Scanner sc = new Scanner(System.in);
      System.out.print("N = ");
      int N = sc.nextInt();
      int[] a = new int[N];
      System.out.println("Array = ");
      for(int i=0 ; i<N ; i++)
      {
         a[i] = sc.nextInt();
         Mp.put(a[i], Mp.getOrDefault(a[i], 0) + 1);

      }
      for(Map.Entry<Integer,Integer> entry : Mp.entrySet())
      {
         if(entry.getValue() % 2 == 0)
         {
            System.out.print(entry.getKey()+" ");
```

```
        }
      }
   }
}
```

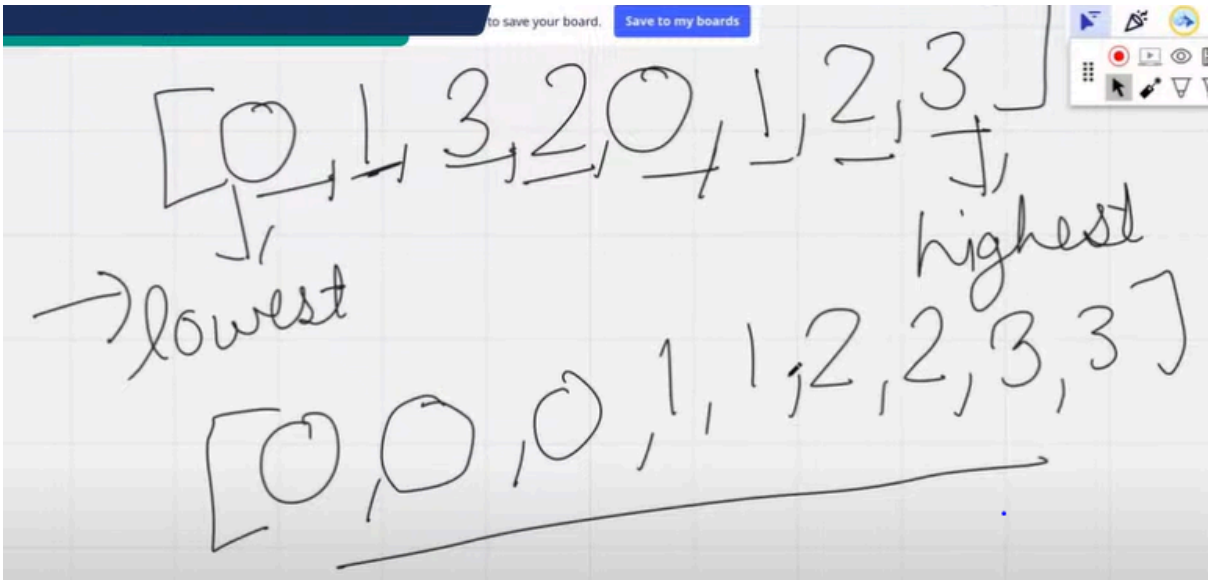
3.

The Metro station 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[0,1,2,3]. 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 3.

a) N is the number of elements of array
b) Next N lines contains value of array

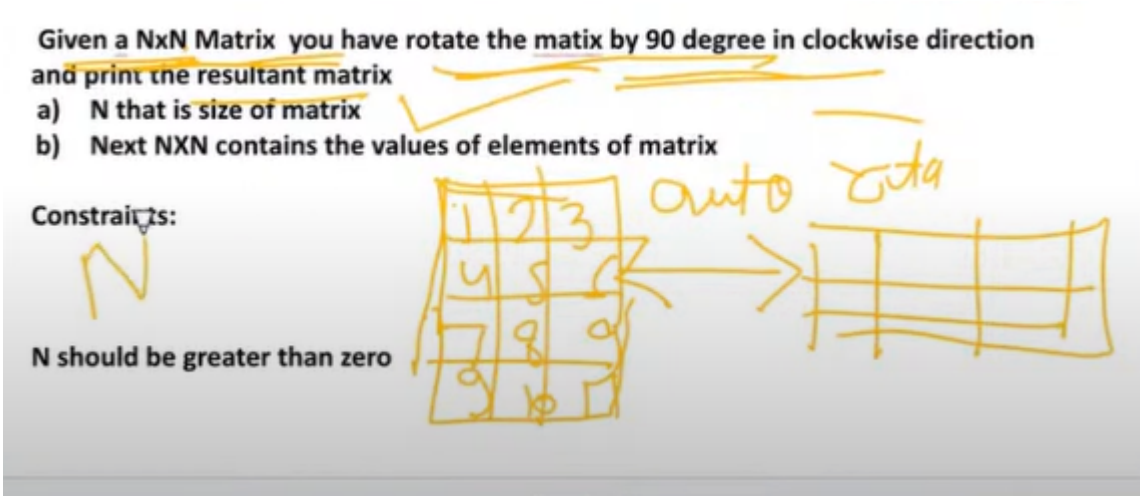
Constraints:

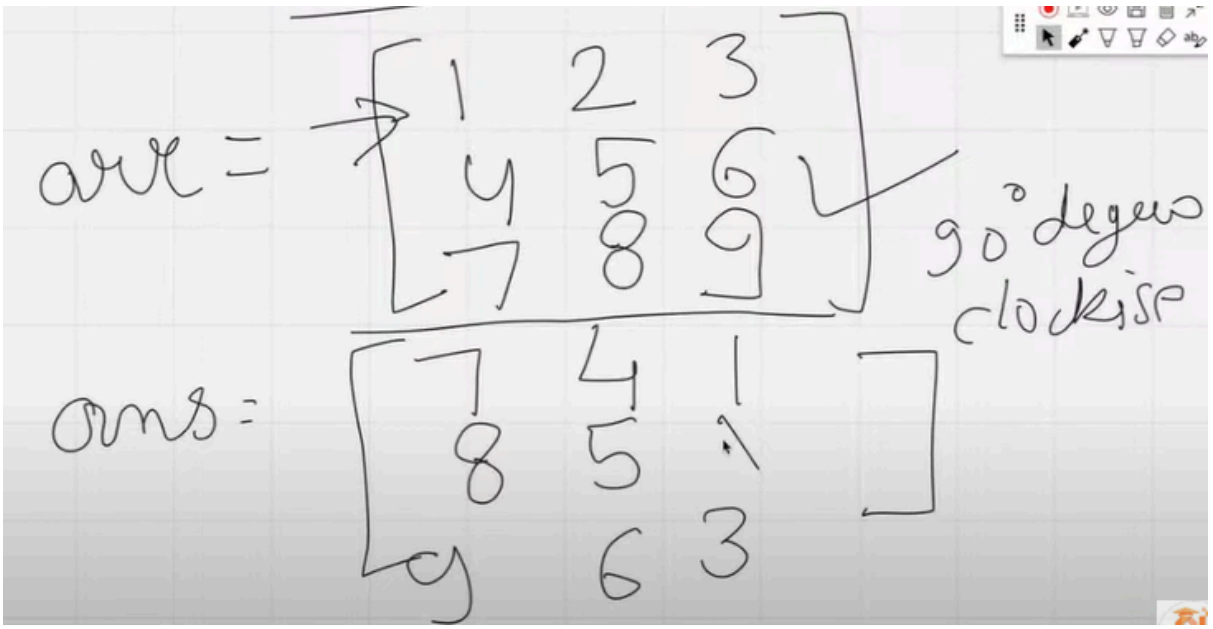
N should be greater than zero
Array contains only the values 0, 1, 2,3




```
import java.util.*;
public class Main
{
        public static void main(String[] args) {
   Scanner sc = new Scanner(System.in);
   int N = sc.nextInt();
   int[] a = new int[N];
   Map<Integer,Integer> Mp = new HashMap<>();
   for(int i=0; i<N ; i++)
   {
```

```
        a[i] = sc.nextInt();
        Mp.put(a[i],Mp.getOrDefault(a[i],0) + 1);
    }
    for(Map.Entry<Integer,Integer> entry : Mp.entrySet())
    {
        if((entry.getKey() == 0) || (entry.getKey() == 1) || (entry.getKey() == 2) || (entry.getKey()
== 3))
        {
            int x = entry.getValue();
            while(x != 0)
            {
                System.out.print(entry.getKey()+" ");
                x--;
            }
        }
    }
  }
}
```






```
import java.util.*;
public class Main{
   public static void main(String[] args)
   {
      Scanner s = new Scanner(System.in);
      int n = s.nextInt();
      int[][] a = new int[n][n];
      for(int i=0 ; i<n ; i++)
      {
         for(int j=0 ; j<n ; j++)
         {
            a[i][j] = s.nextInt();
         }
      }
      for(int i=0 ; i<n ; i++)
      {
      //transpose
         for(int j=i; j<n ; j++)
         {
            int temp = a[i][j];
            a[i][j] = a[j][i];
            a[j][i] = temp;
         }
   }

 for(int i=0 ; i<n ; i++)
   {
        //reverse
         int l = 0;
```

```
            int h = n-1;
            while(l<=h)
        {
            int x = a[i][h];
            a[i][h] = a[i][l];
            a[i][l] = x;
            l++;
            h--;
        }

    }
    System.out.println();

        for(int i=0 ; i<n ; i++)
        {
            for(int j=0 ; j<n ; j++)
            {
                System.out.print(a[i][j]+" ");
            }
                System.out.println();
        }
    }
}
```