https://course.acciojob.com/idle?question=2e44a4b2-25f2-4fba-b95b-302ec00fb872

- EASY
- Max Score: 30 Points

Boolean Matrix Problem

You are given a matrix `Mat` of `m` rows and `n` columns. The matrix is boolean so the elements of the matrix can only be either 0 or 1.

Now, if any row of the matrix contains a 1, then you need to fill that whole row with 1. After doing the mentioned operation, you need to return the modified matrix.

Your task is to complete the function `BooleanMatrixProblem` which takes `m`, `n` and the input array as the parameter and returns the modified matrix as specified in the question.

Input Format

The first line of contains `m` and `n` denoting number of rows and number of columns respectively.

Then next `m` lines contain `n` elements each denoting the elements of the matrix.

Output Format

Return the modified matrix as specified above.

Example 1

Input

```
5 4
1 0 0 0
0 0 0 0
0 1 0 0
0 0 0 0
0 0 0 1
```

Output

```
1 1 1 1
0 0 0 0
1 1 1 1
0 0 0 0
1 1 1 1
```

Explanation

rows = 5 and columns = 4

The given matrix is

```
1 0 0 0
0 0 0 0
0 1 0 0
0 0 0 0
0 0 0 1
```

Evidently, the first row contains a 1 so fill the whole row with 1. The third row also contains a 1 so that row will be filled too. Finally, the last row contains a 1 and therefore it needs to be filled with 1 too.

The final matrix is

```
1 1 1 1
0 0 0 0
1 1 1 1
0 0 0 0
1 1 1 1
```

Example 2

Input

```
2 2
1 0
0 1
```

Output

```
1 1
```

```
1 1
```

Explanation

rows = 2 and columns = 2

The given matrix is

```
1 0
0 1
```

Evidently, the first row contains a 1 so fill the whole row with 1. The second row also contains a 1 so that row will be filled too. The final matrix is

```
1 1
1 1
```

Constraints

1 <= m, n <= 300

Mat[i][j] ∈ {0,1}

Topic Tags

- **2D-Arrays**

My code

```
// in java
import java.util.*;
import java.lang.*;
import java.io.*;

public class Main
```

```
{
    public static void main (String[] args) throws java.lang.Exception
    {
        //your code here
    Scanner s=new Scanner(System.in);
   int n=s.nextInt();
   int m=s.nextInt();
   int arr[][]=new int[n][m];
    for (int i=0;i<n;i++)
      for(int j=0;j<m;j++)
        arr[i][j]=s.nextInt();
  for (int i=0;i<n;i++){
    int c=0;
      for(int j=0;j<m;j++)
        {
          if(arr[i][j]==1) c=1;
        }
    if(c==1)
    for(int j=0;j<m;j++)
        {
          arr[i][j]=1;
        }
  }
      for(int i=0;i<n;i++){
      for(int j=0;j<m;j++)    {
   System.out.print(arr[i][j] +" ");}
      System.out.print("\n");}
```

```
        }
    }
```