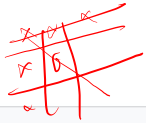
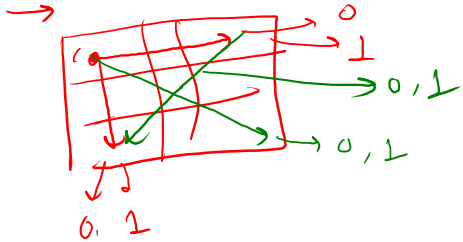
# Tic-Tac-Toe



Problem | Submissions | Leaderboard | Discussions

Given a 3*3 matrix that contains only 0 and 1. There are two players, player 0 and player 1. A player wins if any row or column or left diagonal or right diagonal contains only the value 0 or only the value 1. If any player wins then print “Game ends” otherwise print “Draw”



Sample Input 0

```
3
1 1 0
0 0 1
1 0 1
```

Sample Output 0

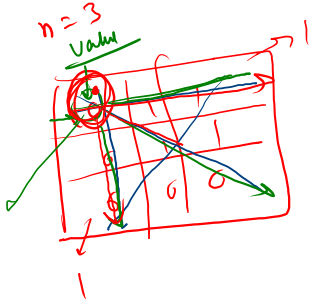
```
Games End
```

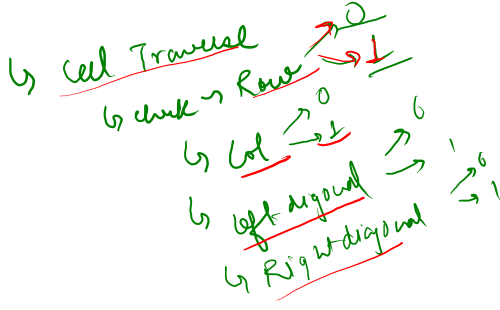
Sample Input 1

```
3
1 1 0
0 0 1
1 0 1
```

Sample Output 1

```
Draw
```





## Sample Input 1

```
3
1 1 0
0 0 1
1 0 1
```

## Sample Output 1

```
Draw
```

Row

```
public static boolean checkRow(int arr[][],int row){
    int n=arr.length;

    int count=0;
    for(int j=0;j<n;j++){
        if(arr[row][j]==0){
            count++;
        }
    }

    if(count==n){
        return true;
    }

    for(int j=0;j<n;j++){
        if(arr[row][j]==1){
            count++;
        }
    }

    if(count==n){
        return true;
    }

    return false;
}
```

Col

```
public static boolean checkCol(int arr[][],int col){
    int n=arr.length;

    int count=0;
    for(int i=0;i<n;i++){
        if(arr[i][col]==0){
            count++;
        }
    }

    if(count==n){
        return true;
    }

    for(int i=0;i<n;i++){
        if(arr[i][col]==1){
            count++;
        }
    }

    if(count==n){
        return true;
    }

    return false;
}
```

```
public static boolean checkRightDiagonal(int arr[][],int row,int col){
    int n=arr.length;

    int count=0;
    for(int i=row,j=col;i<n && j>=0;j--,i++){
        if(arr[i][j]==0){
            count++;
        }
    }

    if(count==n){
        return true;
    }


     count=0;
    for(int i=row,j=col;i<n && j>=0;j--,i++){
        if(arr[i][j]==1){
            count++;
        }
    }

    if(count==n){
        return true;
    }

    return false;
}
}
```

ans 1 = false

ans 2 = false

ans 3 = false

ans 4 = false

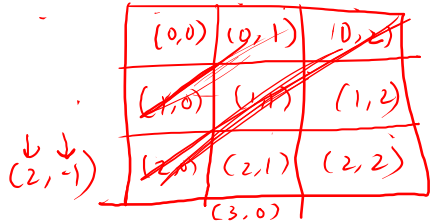
i=0, j=0

i<n && j<n

i++, j++

(0,1) → (1,0)

(0,2) → (1,1) → (2,0)




(2,-1)

j = 0

j = 2

i<n && j>=0 | i++, j--

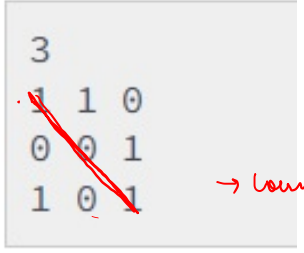
```java
public static boolean checkRightDiagonal(int arr[][],int row,int col){
    int n=arr.length;

    int count=0;
    for(int i=row,j=col;i<n && j>=0;j--,i++){
        if(arr[i][j]==0){
            count++;
        }
    }

    if(count==n){
        return true;
    }


     count=0;
    for(int i=row,j=col;i<n && j>=0;j--,i++){
        if(arr[i][j]==1){
            count++;
        }
    }

    if(count==n){
        return true;
    }

    return false;
}
```

```java
public static boolean checkLeftDiagonal(int arr[][],int row,int col){
    int n=arr.length;

    int count=0;
    for(int i=row,j=col;i<n && j<n;j++,i++){
        if(arr[i][j]==0){
            count++;
        }
    }

    if(count==n){
        return true;
    }


     count=0;
    for(int i=row,j=col;i<n && j<n;j++,i++){
        if(arr[i][j]==1){
            count++;
        }
    }

    if(count==n){
        return true;
    }

    return false;
}
```

```java
public static boolean checkCol(int arr[][],int col){
    int n=arr.length;

    int count=0;
    for(int i=0;i<n;i++){
        if(arr[i][col]==0){
            count++;
        }
    }

    if(count==n){
        return true;
    }
    for(int i=0;i<n;i++){
        if(arr[i][col]==1){
            count++;
        }
    }

    if(count==n){
        return true;
    }

    return false;
}
```

```java
public static boolean checkRow(int arr[][],int row){
    int n=arr.length;

    int count=0;
    for(int j=0;j<n;j++){
        if(arr[row][j]==0){
            count++;
        }
    }

    if(count==n){
        return true;
    }
    for(int j=0;j<n;j++){
        if(arr[row][j]==1){
            count++;
        }
    }

    if(count==n){
        return true;
    }

    return false;
}
```

```java
public static String ticTacToe(int arr[][]){
    int n=arr.length;

    for(int i=0;i<n;i++){
        for(int j=0;j<n;j++){
            boolean ans1=checkRow(arr,i);
            boolean ans2=checkCol(arr,j);
            boolean ans3=checkLeftDiagonal(arr,i,j);
            boolean ans4=checkRightDiagonal(arr,i,j);

            if(ans1==true || ans2==true || ans3==true || ans4==true ){
                return "Games End";
            }
        }
    }

    return "Draw";
}
```



# N Queens Check

Problem | Submissions | Leaderboard | Discussions

Given a n*n chess board that contains 0 and 1. 0 means there is nothing at that place. 1 means there is a queen at that place. Print "N Queens" if no queen attacks the other queen, otherwise print "Danger".

**Sample Input 0**

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

**Sample Output 0**

```
N Queens
```

**Sample Input 1**

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

**Sample Output 1**

```
Danger
```