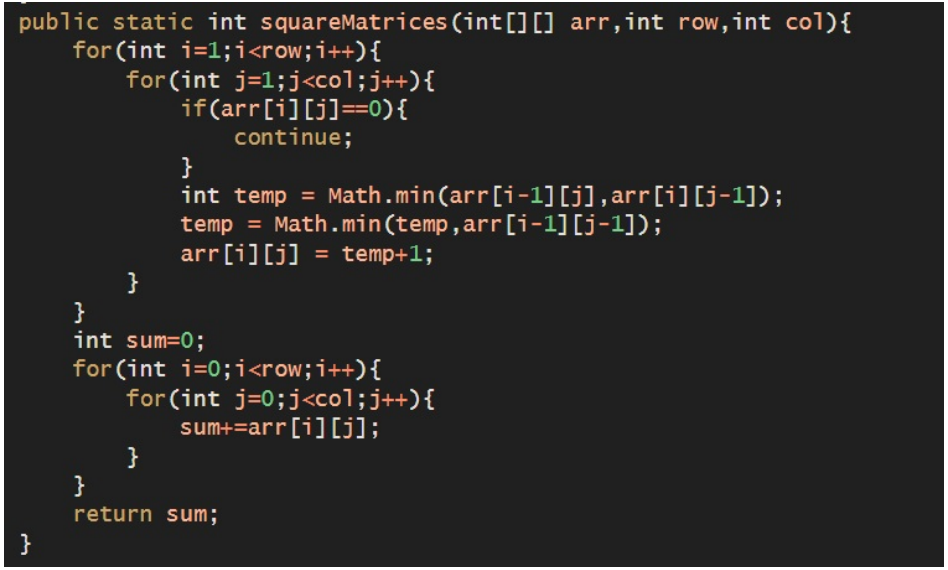
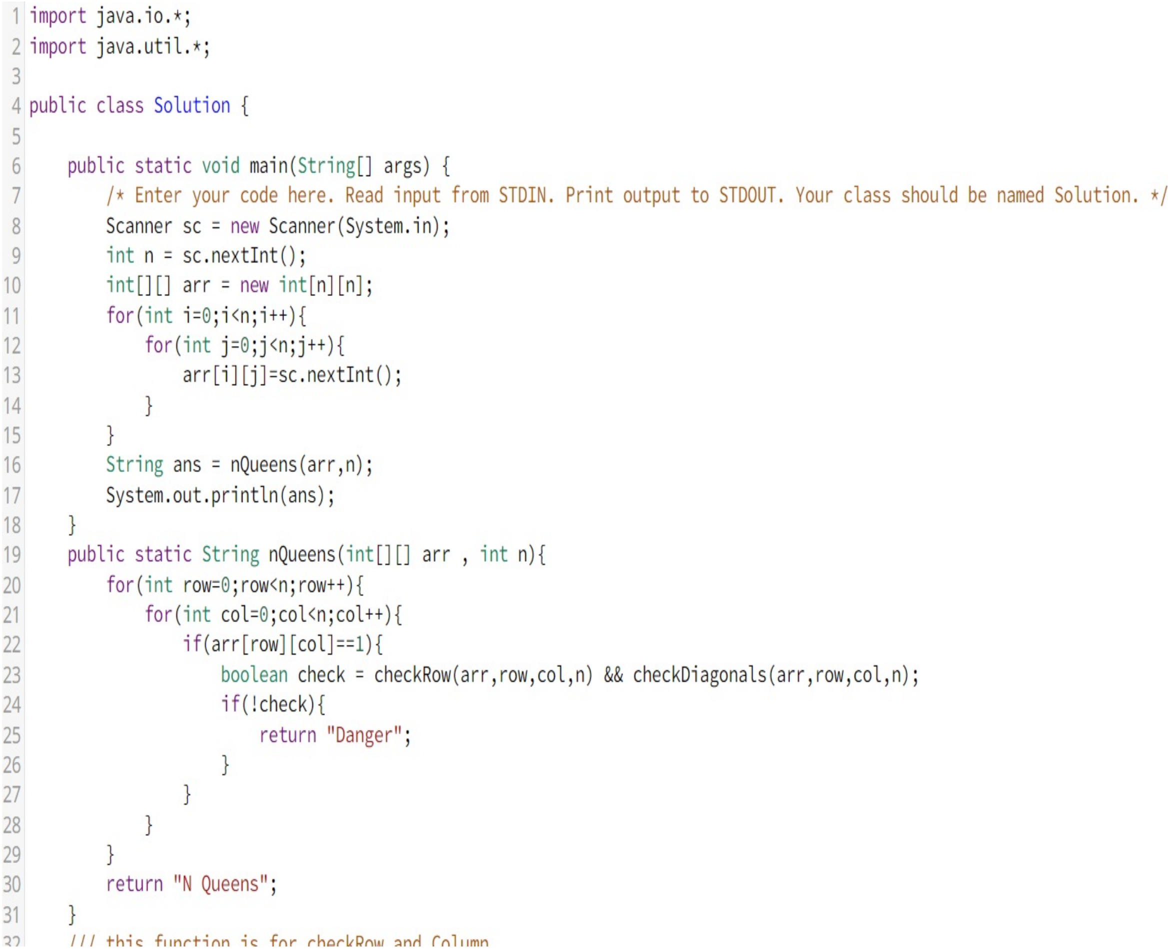
```
3
4
0 1 1 1
1 1 1 1
0 1 1 1
```

```
3
4
0 1 1 1
1 1 2 1
0 1 1 1
```



```java
public static int squareMatrices(int[][] arr,int row,int col){
    for(int i=1;i<row;i++){
        for(int j=1;j<col;j++){
            if(arr[i][j]==0){
                continue;
            }
            int temp = Math.min(arr[i-1][j],arr[i][j-1]);
            temp = Math.min(temp,arr[i-1][j-1]);
            arr[i][j] = temp+1;
        }
    }
    int sum=0;
    for(int i=0;i<row;i++){
        for(int j=0;j<col;j++){
            sum+=arr[i][j];
        }
    }
    return sum;
}
```

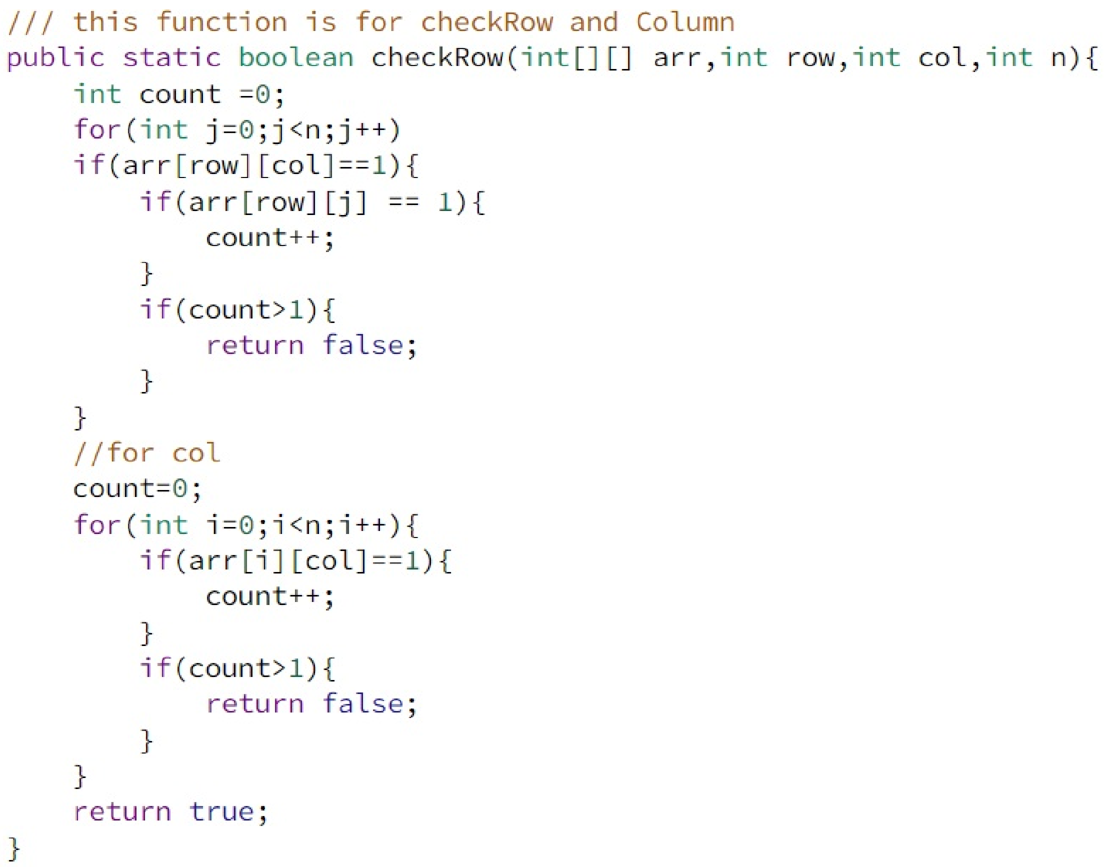
```
4
0 1 0 0
0 0 1 0
1 0 0 0
0 0 1 0
```



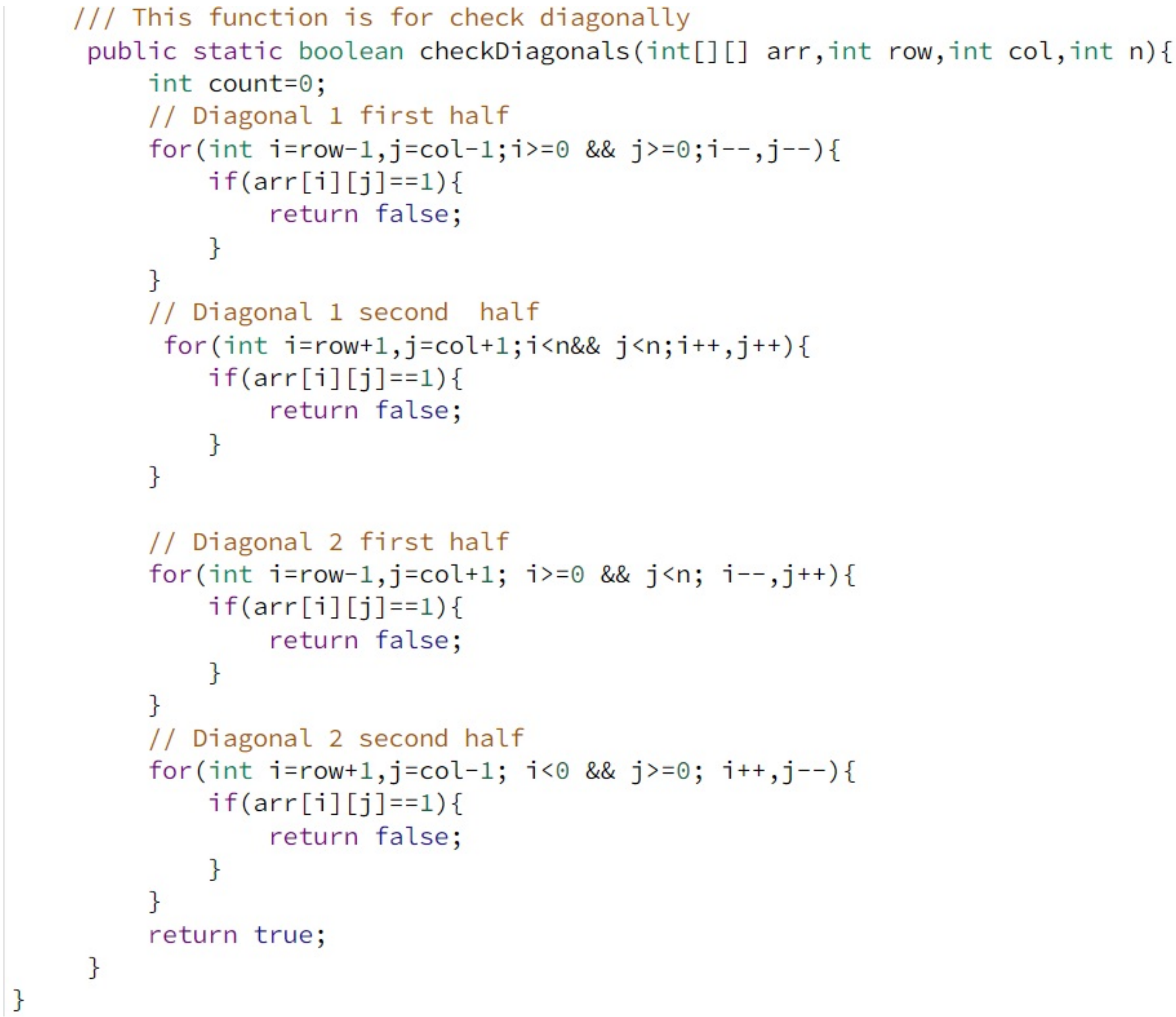
```java
import java.io.*;
import java.util.*;

public class Solution {

    public static void main(String[] args) {
        /* Enter your code here. Read input from STDIN. Print output to STDOUT. Your class should be named Solution. */
        Scanner sc = new Scanner(System.in);
        int n = sc.nextInt();
        int[][] arr = new int[n][n];
        for(int i=0;i<n;i++){
            for(int j=0;j<n;j++){
                arr[i][j]=sc.nextInt();
            }
        }
        String ans = nQueens(arr,n);
        System.out.println(ans);
    }
    public static String nQueens(int[][] arr , int n){
        for(int row=0;row<n;row++){
            for(int col=0;col<n;col++){
                if(arr[row][col]==1){
                    boolean check = checkRow(arr,row,col,n) && checkDiagonals(arr,row,col,n);
                    if(!check){
                        return "Danger";
                    }
                }
            }
        }
        return "N Queens";
    }
    /// this function is for checkRow and Column
```



```
/// this function is for checkRow and Column
public static boolean checkRow(int[][] arr,int row,int col,int n){
    int count =0;
    for(int j=0;j<n;j++)
    if(arr[row][col]==1){
        if(arr[row][j] == 1){
            count++;
        }
        if(count>1){
            return false;
        }
    }
    //for col
    count=0;
    for(int i=0;i<n;i++){
        if(arr[i][col]==1){
            count++;
        }
        if(count>1){
            return false;
        }
    }
    return true;
}
```



```
    /// This function is for check diagonally
     public static boolean checkDiagonals(int[][] arr,int row,int col,int n){
         int count=0;
         // Diagonal 1 first half
         for(int i=row-1,j=col-1;i>=0 && j>=0;i--,j--){
             if(arr[i][j]==1){
                 return false;
             }
         }
         // Diagonal 1 second  half
          for(int i=row+1,j=col+1;i<n&& j<n;i++,j++){
             if(arr[i][j]==1){
                 return false;
             }
         }

         // Diagonal 2 first half
         for(int i=row-1,j=col+1; i>=0 && j<n; i--,j++){
             if(arr[i][j]==1){
                 return false;
             }
         }
         // Diagonal 2 second half
         for(int i=row+1,j=col-1; i<0 && j>=0; i++,j--){
             if(arr[i][j]==1){
                 return false;
             }
         }
         return true;
     }
}
```