矩阵稀疏扫描题目描述：

如果矩阵中的许多系数都为零，那么该矩阵就是稀疏的。对稀疏现象有兴趣是因为它的开发可以带来巨大的计算节省，并且在许多大的实践中都会出现矩阵稀疏的问题。

给定一个矩阵，现在需要逐行和逐列地扫描矩阵，如果某一行或者某一列内，存在连续出现的 0 的个数超过了行宽或者列宽的一半$/lfloorW/2/rfloor$（地板除），则认为该行或者该列是稀疏的。

扫描给定的矩阵，输出稀疏的行数和列数。

输入描述：

第一行输入为 M 和 N，表示矩阵的大小

$M/timesN$，

$0<M/leq100,0<N/leq100$。

接下来 M 行输入为矩阵的成员，每行 N 个成员，矩阵成员都是有符号整数，范围-32,768 到 32,767。

输出描述：

输出两行，第一行表示稀疏行的个数，第二行表示稀疏列的个数。

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

// 注意类名必须为 Main, 不要有任何 package xxx 信息
public class Main {
    public static void main(String[] args) {
        Scanner in = new Scanner(System.in);
        // 注意 hasNext 和 hasNextLine 的区别
        while (in.hasNextInt()) {
            // 注意 while 处理多个 case
            int a = in.nextInt();
            int b = in.nextInt();
            int[] row = new int[a];
            int[] col = new int[b];
```

```
int[][] rowAndCol = new int[a][b];
int r2 = b / 2; //行的地板除
int c2 = a / 2; //列的地板除

for (int i = 0; i < a; i++) {
	for (int j = 0; j < b; j++) {
		rowAndCol[i][j] = in.nextInt();
	}
}
HashMap<Integer, Integer> rowMap = new HashMap<Integer, Integer>();
for (int i = 0; i < a; i++) {
	for (int j = 0; j < b; j++) {
		if (!rowMap.containsKey(i)) {
			if (rowAndCol[i][j] == 0) {
				rowMap.put(i, 1);
			}
		} else {
			if (rowAndCol[i][j] == 0) {
				rowMap.put(i, rowMap.get(i) + 1);
			}
		}
	}
}
HashMap<Integer, Integer> colMap = new HashMap<Integer, Integer>();
for (int i = 0; i < b; i++) {
	for (int j = 0; j < a; j++) {
		if (!colMap.containsKey(i)) {
			if (rowAndCol[j][i] == 0) {
				colMap.put(i, 1);
			}
		} else {
			if (rowAndCol[j][i] == 0) {
				colMap.put(i, colMap.get(i) + 1);
			}
		}
	}
}
int rowCount = 0;
for (Integer r : rowMap.keySet()) {
	if (rowMap.get(r) >= r2) {
		rowCount++;
	}

}
```

```
                int colCount = 0;
                for (Integer c : colMap.keySet()) {
                        if (colMap.get(c) >= c2) {
                                colCount++;
                        }
                }
                System.out.println(rowCount);
                System.out.println(colCount);
                break;
        }
    }
}
```