

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（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，。

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。

补充说明：

```
示例1
输入：3 3
      1 0 0
      0 1 0
      0 0 1
输出：3
      3
说明：给定的3 × 3矩阵里，每一行和每一列内都存在2个0，行宽3，列宽3， $\lfloor 3/2 \rfloor = 1$ ，因此稀疏行有3个，稀疏列有3个。

示例2
输入：5 3
      -1 0 1
      0 0 0
      -1 0 0
      0 -1 0
      0 0 0
      10
输出：5
      3
说明：给定的5 × 3矩阵，每行里面0的个数大于等于1表示稀疏行，每列里面0的个数大于等于2表示稀疏列，所以有5个稀疏行，3个稀疏列。
```

```
import java.util.Scanner;
```

```
public class Main {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        int n = sc.nextInt();
        int m = sc.nextInt();
        int[][] grid = new int[n][m];
        for (int i = 0; i < n; ++i) {
            for (int j = 0; j < m; ++j) {
                grid[i][j] = sc.nextInt();
            }
        }
        int hang = 0;
        for (int i = 0; i < n; ++i) {
            int now = 0;
            for (int j = 0; j < m; ++j) {
```

```

        if (grid[i][j] == 0) {
            now += 1;
        }
    }
    if (now >= m / 2) {
        hang += 1;
    }
}
System.out.println(hang);

int lie = 0;
for (int j = 0; j < m; ++j) {
    int now = 0;
    for (int i = 0; i < n; ++i) {
        if (grid[i][j] == 0) {
            now += 1;
        }
    }
    if (now >= n / 2) {
        lie += 1;
    }
}
System.out.println(lie);
}
}

```