Today Mr. Ed lent his phone to Ethan. He didn't lock his phone with password because he thought Ethan is a reliable guy, but he wrote "I'm gay" on Ed's timeline. Obviously Mr. Ed is very pissed.

Due to the recent betrayal to his trust, Mr. Ed wants to lock his phone with a pattern. You know what a phone unlock pattern is, right? The lock screen of a phone consists in a grid of n rows and m columns of dots (or circles). Mr. Ed can choose any sequence of dots $S = d_1, d_2, \ldots, d_k$ as the phone unlock pattern; this means that, in order to unlock the phone, Ed needs to trace a path starting in d_1 and ending in d_k that passes along every dot in S in exactly the same order.

There are two restrictions to the sequence of dots S: first, every dot d_i must appear at most one time in the sequence and; for every two consecutive dots in S, let's call them d_i and d_{i+1}, the straight line from dot d_i to d_{i+1} must not pass by any dot not previously visited in the sequence. For example, in a 1×3 grid, the sequence $S = 1, 3, 2$ is not possible since the straight line from 1 to 3 passes through 2 (not visited yet).
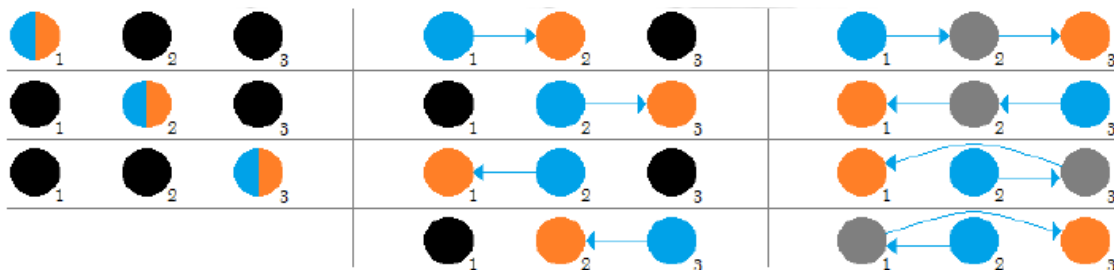

Figure: Possible unlock patterns for a 1×3 grid using 1 dot (left), 2 dots (middle) and 3 dots (right).

The above figure illustrates the 11 possible unlock patterns for a lock screen of 1×3 dot grid; blue dots represent the starting dot of the pattern and orange dots represent the respective end dot.

Please help Mr. Ed counting the number of unlock patterns he could use to lock his phone.

Input

The input will contain several test cases. Each test case consists of a single line with two integer numbers $1 \leq n \leq 3$ and $1 \leq m \leq 3$: the number of rows and columns in the lock screen grid. The last test case is followed by a single line containing 2 zeroes, which should not be processed.

Output

For each test case, print the number of possible unlock patterns Mr. Ed could use (see format below).

Sample Input

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

Sample Output

```
Case #1: 4
Case #2: 11
```