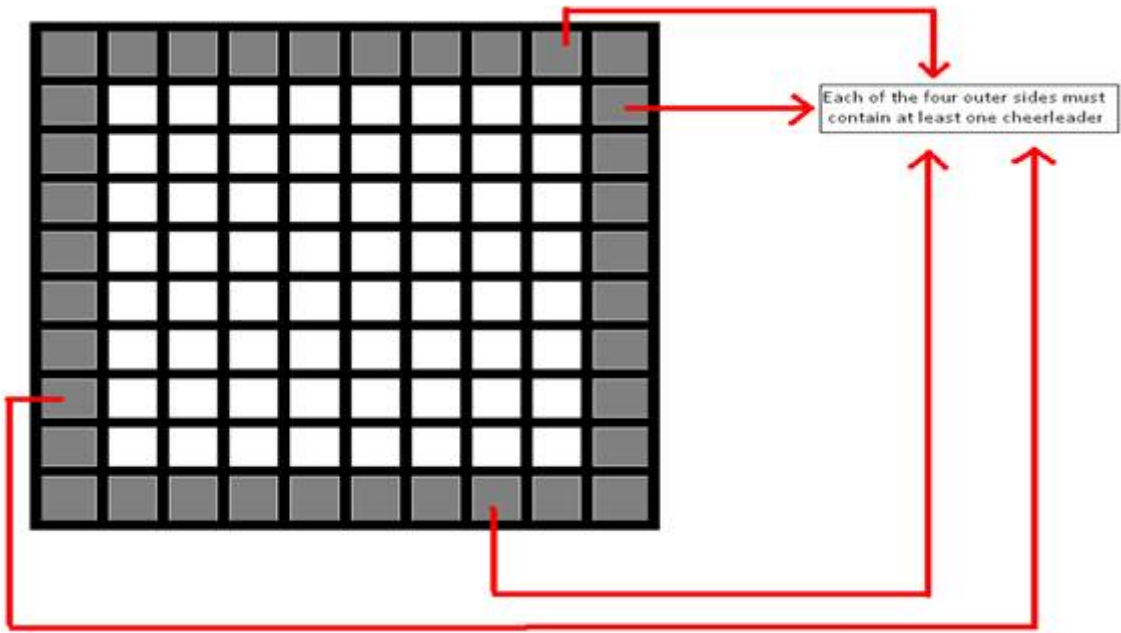# 11806   Cheerleaders

In most professional sporting events, cheerleaders play a major role in entertaining the spectators. Their roles are substantial during breaks and prior to start of play. The world cup soccer is no exception. Usually the cheerleaders form a group and perform at the centre of the field. In addition to this group, some of them are placed outside the side line so they are closer to the spectators. The organizers would like to ensure that at least one cheerleader is located on each of the four sides. For this problem, we will model the playing ground as an $M \times N$ rectangular grid. The constraints for placing cheerleaders are described below:

- There should be at least one cheerleader on each of the four sides. Note that, placing a cheerleader on a corner cell would cover two sides simultaneously.

- There can be at most one cheerleader in a cell.

- All the cheerleaders available must be assigned to a cell. That is, none of them can be left out.

Each of the four outer sides must contain at least one cheerleader

The organizers would like to know, how many ways they can place the cheerleaders while maintaining the above constraints. Two placements are different, if there is at least one cell which contains a cheerleader in one of the placement but not in the other.

Input

The first line of input contains a positive integer $T \leq 50$, which denotes the number of test cases. T lines then follow each describing one test case. Each case consists of three nonnegative integers, $2 \leq M$, $N \leq 20$ and $K \leq 500$. Here M is the number of rows and N is the number of columns in the grid. K denotes the number of cheerleaders that must be assigned to the cells in the grid.

Output

For each case of input, there will be one line of output. It will first contain the case number followed by the number of ways to place the cheerleaders as described earlier. Look at the sample output for exact formatting. Note that, the numbers can be arbitrarily large. Therefore you must output the answers *modulo* 1000007.

Sample Input

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

Sample Output

```
Case 1: 0
Case 2: 2
```