UVa Online Judge

11194 Stone Grid

In the country of Byteland, there is a grid of R rows and C columns. In this grid, most of the cells contain some number of marbles. At each step, the Bytelandian people perform one of the following operations:

- They can select any pair of adjacent cells and place one stone into each of the two cells.
- They can select any pair of adjacent cells and take one stone from each of the two cells. None of the two cells are allowed to be empty for this operation to take place.

Some of the cells are evil and may not be used in either of the above operations. The Bytelandian people are given a grid with a number of stones in the cells. They want to rearrange the stones using the two operations. They are not allowed to waste stones (which will be used for another purpose later), so the do not want the final stone arrangement to have more than L stones. For the same reason, they do not want the final arrangement to have more stones than the initial grid configuration does. Now they wonder how many different final stone arrangements in the grid they can reach. Two arrangements are different if there exists a cell, such that the two arrangements differ in the number of stones in that cell.

Input

The first line of input gives the number of cases, T ($T \leq 40$). T test cases follow.

Each one starts with 3 integers - R, C ($1 \leq R, C \leq 20$) and L ($0 \leq L \leq 1000$). Each of the next R lines contains C integers. The j-th integer in the i-th row denotes the number of initial stones in the cell (i, j). If the integer is '`-1`' then the cell is evil. The number of initial stones in each cell does not exceed 10.

There is a blank line between two consecutive test cases in the input file.

Output

For each test case, output a single integer, giving the total number of final arrangements the Bytelandians can reach. The result may be very big, so output only its remainder after division by 10007, that is $result\%10007$.

Sample Input

```
4

1 2 2
1 1

1 2 6
0 1

2 2 2
1 1
1 1

3 2 4
1 1
```

```
-1 -1
1 1
```

Sample Output

```
2
1
5
4
```