A rectangular board of M columns and N rows is drawn on a piece of graph paper. Some cells of the board are empty; the others are filled with debris. Each cell of the board has two coordinates; if a cell's coordinates are (i, j), this cell is in the i-th column from the left and in jth row from the top. Therefore, the upper left cell of the board is $(1,1)$, and the lower right cell is (M, N).

Consider two cells A and B. A *path* between A and B is a sequence of cells $A = c_0, \ldots, c_k = B$ such that for each i, $1 \leq i \leq k$, the cells c_{i-1} and c_i have a common side. The number k is called the *length* of the path. Let (x, y) be the coordinates of A minus the coordinates of B. The cells A and B are called *connected* if there is a path between A and B of length $|x| + |y|$.

You have an automobile standing in the upper left cell. Your automobile has a speed vector with integer coordinates which is originally $(0,0)$. The automobile can make a move as follows:

- The speed vector changes. Each coordinate of the speed can change by at most 1. Moveover, the X coordinate of the speed vector must be between 0 and P and the Y coordinate of the speed vector must be between $-Q$ and Q, where P and Q are given. If the X coordinate of the speed vector is 0, then the Y coordinate of the speed vector must be greater than 0.

- The speed vector is added to the coordinates, thereby changing the location. The new location must remain on the board, and the previous and current locations must be connected.

Your task is to find the minimal number of moves which will take the automobile to the lower right cell.

Input

The first line of the input contains the number of the test cases. which is at most 20. The descriptions of the test cases follow. The first line of a testcase contains four integers M, N, P, and Q ($2 \leq M, N \leq 250, MN \leq 500, 1 \leq P, Q \leq 10$) separated by spaces. Each of the next N lines contains M numbers describing the board. The j-th number in the $i + 1$-st line is 1 if the cell (i, j) is occupied with debris and 0 if this cell is empty. It is guaranteed that the upper left and lower right cells are empty. The test cases are separated by blank lines.

Output

For each test case in the input, output 'Impossible' (without quotes) if it is impossible to get from the upper left cell to the lower right. Otherwise, output the minimal possible number of moves to get from the upper left cell to the lower right, followed by the sequence of the coordinates of the cells on the optimal path separated by spaces and/or end of line characters. If there are several optimal solutions, output any of them. Print a blank line between test cases.

Sample Input

```
2

7 3 3 3
0 0 0 0 0 0 0
0 1 1 1 1 1 0
0 0 0 0 0 0 0

5 4 3 3
0 1 0 0 0
0 1 0 1 0
0 1 0 1 0
0 0 0 1 0
```

Sample Output

```
4
1 1
1 2
2 3
4 3
7 3

Impossible
```