

4464 May the Right-Force be with you

In a sensational scene in the 8th installment of the *Voldemort* book series, the robot *Weighd* was attacked and its rotating top and wings were severely damaged. *Weighd* cannot rotate its famous top and can only fly forward or make a right-turn. However, the brave *Weighd* must still navigate a magically created series of mazes to deliver its precious message to *Ttoper*.

Weighd enters each maze at the top-left square of the maze, takes one unit of time to travel between adjacent squares in the magic maze (forward or right), and cannot stay in one square to rotate itself, and thus change direction, for fear of evil jinx. Your task is to write a program to calculate the smallest number of squares that *Weighd* must travel to reach a certain square in each maze. Do not you worry about how? *Weighd* will feel its correct way through the force. Examples are:

| | | | | | | | | | | | | | |
|-------|---|---|---|---|---|---|---|---|---|---|---|----------|--|
| start | f | f | f | f | r | | | | | | | | |
| | | | | | f | | | | | | | | |
| | | | | r | f | f | f | f | f | f | f | r | |
| | | | | r | r | | | | | | | X | |
| | | | | | | | | | | | | | |
| | | | | | | | | | | | | | |

“f” means forward ,“r” means a right-turn and “?” means no where to go.
Note that the green square is crossed twice by Weighd.

| | | | | | | | | | | | | | |
|-------|---|---|---|---|-----|--|--|--|--|--|--|----------|--|
| start | f | f | f | f | r | | | | | | | | |
| | | | | | f | | | | | | | | |
| ? | f | f | f | f | r/f | | | | | | | | |
| | | | | | f | | | | | | | X | |
| | | | | | ? | | | | | | | | |

Weighd can reach the location marked “X” of the top maze in 19 steps, but it cannot reach the marked location in the bottom maze at all.

Input

Input consists of multiple mazes. Each maze description starts with two integers, on a separate line, that represent its dimensions. The first integer N ($1 \leq N \leq 1000$) represents the number of rows and the second M ($1 \leq M \leq 1000$) represents the number of columns. The last maze is followed by a line containing two zeros that indicate the end of the input data and should not be processed as a valid situation.

The second line contains two integers, C ($1 \leq C \leq N$) and R ($1 \leq R \leq M$), that represent the column number and the row number of the square where *Weighd* must reach in the maze. Consecutive integers are separated by a blank space.

Each of the following N lines contains a sequence of ‘0’s and ‘1’s. The sequence is M characters long. The sequence does not contains blank spaces. A value one (1) represents a wall in the maze (that is, a square into which *Weighd* cannot fly).

Output

Output consists of one line for each maze. It will be in one of the following two formats:

1. an integer that represents the number of steps to be taken, inside the maze, by *Weighd* to reach its destination.
2. The string 'N000!', if no path can be found.

Sample Input

```
6 13
13 4
0000001111000
0111101111000
0000000000000
0000001111110
0000001111110
0000000000000
6 13
13 4
0000001111000
1111101111000
0000000000000
0001101111110
0001101111110
0001100000000
0 0
```

Sample Output

```
19
N000!
```