

# Walking Robot

## Walking Robot

Leonard is a student of robotics, and his teacher, Dr. Cooper, asked his class to build a robot to interact with the environment and change its behavior upon certain events. The environment, behavior changes and the events are up to the student, so Leonard decided to make a robot that walks in a grid with a set of movements. There are downloading spots in the grid in which the robot can learn new movements. The task of the robot is to go from one point of the grid to another in as fewest steps as possible.

The grid is composed of  $M$  rows and  $N$  columns of squares. The bottom-left square is position  $(0,0)$  and the top-right square is position  $(N-1,M-1)$ . A movement is defined as a tuple  $(X,Y)$ . Where  $X$  denotes the movement on the  $x$  axis and  $Y$  on the  $y$  axis. A movement is considered as 1 step. The robot can not make a movement that will end outside the boundaries of the grid.

There are  $K$  possible movements for the robot to learn. The robot starts with  $T$  movements on its memory. If the robot is in a downloading spot, he can choose to learn the movement available at that spot, this takes 1 step. The robot can not learn new movements while he is moving. The robot starts at position  $(0,0)$  and should move to position  $(N-1,M-1)$ .

## Input

The first line has four integers  $M, N$  ( $2 \leq M, N \leq 15$ ),  $K$  ( $1 \leq K \leq 12$ ) and  $T$  ( $0 \leq T \leq K$ ). After this line there will be  $K$  lines. The  $K$ th line contains two integers  $(X,Y)$  ( $-10 \leq X, Y \leq 10$ ) which describe the  $K$ th movement. Movements are indexed from 1 to  $K$  inclusive. Afterwards there will be a line with  $T$  numbers separated by spaces, which describe the movements that are initially known by the robot. Then there is the description of the grid: there will be  $M$  lines with  $N$  integers each. 0 indicates a free square, a positive number means there is a downloading spot for the movement with that number index. The numbers of the grid will always be between 0 and  $K$  inclusive. The last case is followed by a line containing four zeros. This case should not be processed.

## Output

For each test case print a single line with "Case # $X$ :  $S$ " where  $X$  is the number of the test case (starting from 1) and  $S$  is the lowest number of steps to achieve the destination. If it is not possible for the robot to achieve the destination, print -1 instead of the number of steps.

## Example

### Input:

```
4 3 2 1
1 0
0 1
1
0 0 0
0 0 0
```

0 0 0  
0 0 2  
5 5 4 1  
2 1  
-1 1  
-1 0  
-1 -2  
1  
0 4 3 0 0  
0 0 0 0 0  
0 0 0 0 2  
0 0 0 0 0  
0 0 0 0 0  
5 5 4 1  
2 1  
-1 1  
-1 0  
-1 -2  
1  
4 0 3 0 0  
0 0 0 0 0  
0 0 0 0 2  
0 0 0 0 0  
0 0 0 0 0  
7 7 3 0  
1 1  
2 2  
4 4  
0 0 0 0 0 0 0  
0 0 0 0 0 0 0  
0 0 0 0 0 0 0  
0 0 0 0 0 0 0  
0 0 3 0 0 0 0  
0 2 0 0 0 0 0  
1 0 0 0 0 0 0  
0 0 0 0

**Output:**

Case #1: 6  
Case #2: 11  
Case #3: -1  
Case #4: 5