

# Number of Optimal Pac Man Paths

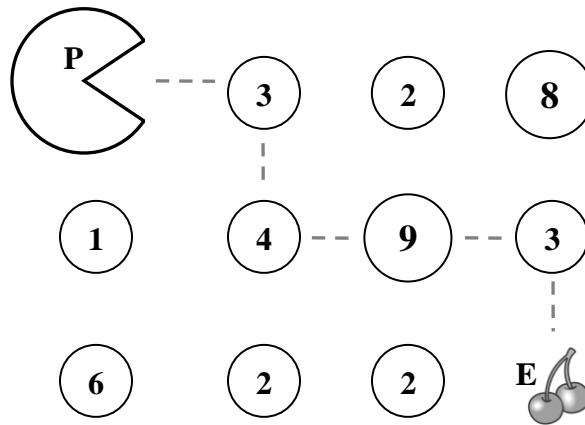
*Filename: pacman.java*

*Time Limit: 5 seconds (per input case)*

*Standard Input, Standard Output*

A few years ago you wrote an app for your friend's new Phone, the newPhone. Since you grew up on Pac Man, you wanted to write a simplified version of the game. In this game, the board was a rectangular grid and Pac Man started at the upper left-hand corner. His goal was to get to the lower right-hand corner. He always moved one square to the right or one square down. Each square he went to had a "goody" that's worth a particular amount of points. Your score was simply the sum of the scores of the goodies in each square you visited.

For example, if the game board looked like this (P indicates Pac Man's starting location, and E indicates his ending location):



then Pac Man's optimal strategy would be to move right, down, right, right again, then down to yield a score of  $3 + 4 + 9 + 3 = 19$ .

Upon reflecting on that first game, you now you realize that there might be more than one path that yields the maximum possible score for some input boards. You are curious just how many optimal paths a board might have. Furthermore, of those paths, you're also curious to determine the first alphabetic path that yields a maximum possible score. For the purposes of this problem, we describe a path as a string consisting only of letters 'D' or 'R', in the order that Pac Man moves. The string "RDRRD" represents the path shown above.

Write a program to satisfy your curiosity!

## The Problem

Given a game board, determine the maximum possible score for Pac Man, the number of different paths that achieve that score and the first alphabetic path as described above that yields the maximum possible score. Two paths are considered different if one path contains a square that the other path does not contain.

### Input

The first line of input will contain two positive integers,  $r$  ( $0 < r \leq 1000$ ),  $c$  ( $0 < c \leq 1000$ ) and  $r+c > 2$ , representing the number of rows and columns for this game board. (The example above has three rows and four columns.) Each of the following  $r$  input lines will contain  $c$  tokens, representing the contents of that row. The first item on the first of these lines will be the character 'P', representing Pac Man's original location and the last item on the last line will be the character 'E', representing Pac Man's goal location. The rest of the items will be positive integers less than or equal to 1000. Items will be separated by a single space on each line.

### Output

Output two lines. The first line of output should contain two space separated integers: the maximum score Pac-Man can achieve and the number of different paths that achieve that score. Since the latter number can be very large, please print the result of this number modulo  $10^9 + 7$ . The second line of output should contain the string describing the first alphabetic path that achieves the maximum score for Pac-Man.

### Samples

Input	Output
3 4 P 2 4 3 2 9 6 9 13 3 8 E	26 3 DDRRR
1 8 P 123 67 13 19 12 200 E	434 1 RRRRRRR
1000 1000 P 1000 1000 ... 1000 1000 1000 ... 1000 ... 1000 1000 E	1997000 965601742 DD...DR..R (Note full output posted in file pacman03.out.)
6 4 P 1 2 3 1 5 6 7 2 5 10 11 3 7 11 15 20 20 20 1 20 20 20 E	86 3 DDDDRRR

**Note: Sample 3 is posted and has 1000 for every goody on a 1000 by 1000 grid.**

### Grading Notes

Since this is the last assignment, to speed up the grading for the TAs, more points (75) will be awarded for correct output. In particular, there will be 25 input cases, and each piece of output for each case will be worth 1 point. ***There are no exceptions to how these 75 points will be awarded, so TAs will not be debugging code at all to try to take off for individual mistakes. This is simply a warning to be very, very, very careful when testing.***