**18. Abandoned Mineshaft**

# Program Name: Mineshaft.java Input File: mineshaft.dat

Oh no! You have fallen into an old abandoned mineshaft! In order to escape, you will have to dig your way through the collapsed walls to find the exit. Unfortunately, your trusty rusty shovel only has a limited number of uses (durability) and can only dig through certain ‘breakable’ walls.

The following characters will be used to describe the mineshaft:

* . - Denotes a clear space
* # - Denotes an unbreakable wall
* % - Denotes a breakable wall
* S - Denotes the starting space
* E - Denotes the exit

Your goal is to find the quickest path to the exit where each movement in the four cardinal directions takes 1 second, each movement down a floor takes 2 seconds, each movement up a floor takes 3 seconds, and breaking any wall takes 3 seconds.

Remember that you can break walls that are directly below or above you.

**Input**

The first line of input is the number of test cases.

For each test case, the first line will consist of four integers in the form f r c a, where f is the number of floors (0 < f < 10), r is the number of rows (0 < r < 100), c is the number of columns (0 < c < 100), and a is the durability of the shovel (0 < a < 100). The next f floors will consist of r lines with c characters on each line. There will be no empty spaces between cases or between lines.

**Output**

The output should be one line per test case in the format # SECONDS where # indicates the number of seconds it took to escape the mineshaft. Output DEAD if there is no way to escape.

**Example Input File**

3

2 2 2 1

S%

#.

##

#E

2 1 1 50

E

S

1 4 4 4

S###  
####  
####  
###E

**Example Output to Screen**

7 SECONDS

3 SECONDS

DEAD