Target Point

Bibi really liked the sniper game, but the rules in this game are a bit unique from the other games. The rules are described below:

1. Spreader (+) had 4 guns set up to all inputs directions(up, down, left, right) which would fire bullets to 3 directions except the triggered side automatically as they were hit by bullets.

2. Wall (#) stops every bullet that comes to it.

3. Start point (S) is the position where Bibi shot the first bullet, bullets can go through this place. For further clarification on the direction of Bibi's shot at start, will be explained further in the Input Section

4. Target point (X) is what Bibi want to shoot.

Bibi really wanted to know how many targets points to be shot when she shot a bullet in a certain direction from the start point(S).

For this problem please make a function:

"int getResult(char map[105][105],int position_x,int position_y,int direction)".
Which gives results, the number of target points that shot by Bibi. You are encouraged to use recursive techniques to solve this problem.

DO NOT include any built-in C/C++ function, except for "stdio.h" function.

Format Input

The first line contains *T*, the number of test cases.

Each test case consists of several inputs:

The 1st line contains N (Row) and M (Column) the size of the map.

The next N-line and M-column contain the map of the game.

The map consists of 5 characters namely wall (#), start point (S), target point (X), spreader (+), and blank (.).

The next line is string *D*, the direction of Bibi's first shot from the start. The direction is exactly "UP", "DOWN", "LEFT" or "RIGHT" without quotes.

Format Output

For each test case output "Case #X: Y" where X is the case number, followed by Y the number of target points that shot by Bibi.
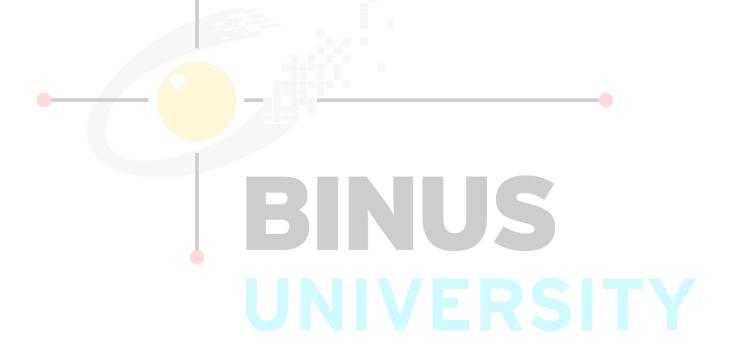
Constraints

$1 \leq T \leq 100$
$3 \leq N, M \leq 100$

Sample Input	Sample Output
3	Case #1: 3
9 9	Case #2: 1
########	Case #3: 0
#.......#	
#...S...#	
#.......#	
#.......#	
X...+...X	
#.......#	
#.......#	
####X####	
DOWN	
11 11	
##########	
#....S...+#	
#.........#	
#.........#	
#.........X	
X........+#	
#.........#	
#.........#	
#.........#	
#.........#	
##########	
RIGHT	
11 9	
########	
#...S...#	
#.......#	
#.......#	
#.......#	
X.......X	
#.......#	
#.......#	

```
#...+...#
#...+...#
########
DOWN
```