Water Drainage

You decide to give up computer science, and instead go into environmental engineering. Luckily, your computer science skills will come in handy! Your first job is to deal with modeling the water run-off – or drainage – in a basin area. Given a representation of the area to model, your task is to determine how far the water will flow.

The land will be represented by height map, which is a two-dimensional grid of heights. Each grid will have r rows and c columns. Each grid location – or grid cell – will have an integer height elevation. Because you will be dealing with multiple grids, each one will have a title as well.

Your task is to figure out the *longest* sequence of grid locations that water can flow between. Water will flow from a higher elevation to a lower elevation. For the purposes of this problem, water will *never* flow from a given elevation to the *same* elevation, nor will it flow uphill. Futhermore, water can only flow from one grid cell to an *adjacent* cell (adjacent cells are above, below, left, and right; not diagonal!).

As an example, consider the following 5x5 grid. Note that the input in this example is justified to help illustrate the grid; there will only be one space between heights in the actual input.

```
66 78 41  3 77
 4 90 41  8 68
12 11 29 24 53
 0 51 58  9 28
97 99 96 58 92
```

There are many such valid drainage paths in this grid. One starts in the second cell of the second row, and flows from 90-78-41-3. Note that 90-41-41-3 is not a valid drainage flow, as water is not always flowing downhill (41-41 is *not* downhill). The longest drainage path in this example is of length 7, and flows from the 99 in the bottom row to the 3 in the top row; the full path is 99-96-58-29-24-8-3.

You could try to solve this problem using top-down recursion or with a brute-force solution, but the running time will be too long. So for this homework, you must use one of the approaches we've studied in this unit: recursion with memoization, or bottom-up dynamic programming. Think carefully about the problem and decide which approach is the better choice.

Input

The first line of the input will be the number of test cases. There will not be more than 100 test cases provided.

Each test case will start with three values space-separated on a single line: the title string of the test case (all letters and numbers; no punctuation or whitespace) followed by r and c, the number of rows and columns, respectively. The following r lines will contain c numbers each, defining the heights of the grid. Both r and c will be positive integers not greater than 100. Each number in the grid itself will be an integer height ranging from 0 to 100.

Output

For each test case, your output should contain a single line that contains the test case title, a colon, a space, and the length of the longest drainage run.

Sample Input

```
4
Charlottesville 5 5
66 78 41 3 77
4 90 41 8 68
12 11 29 24 53
0 51 58 9 28
97 99 96 58 92
Richmond 3 3
1 1 1
1 1 1
1 1 1
WashingtonDC 5 5
10 81 28 2 49
64 59 61 85 82
77 14 81 6 76
37 86 99 11 92
85 95 78 13 57
Wintergreen 5 5
1 2 3 4 5
10 9 8 7 6
11 12 13 14 15
20 19 18 17 16
21 22 23 24 25
```

Sample Output

```
Charlottesville:  7
Richmond:  1
WashingtonDC: 5
Wintergreen:  25
```