Shortest Path Home

Filename: `home`

You are excited that you've finally learned how to calculate the shortest distances on a map in your Computer Science II class. Now, you want to see if you can apply that knowledge to help you save time.

The Problem:

Given a list of intersection points, and the distances between each of those intersection points (when riding your bike), determine the shortest distance and path from the intersection designated as your house to another desired intersection. If there are multiple shortest paths, you must take the one that has the fewest edges. You are guaranteed that none of the input maps will have multiple shortest paths with the minimum number of edges.

The Input:

There will be maps (test cases) in the input file. The first input line contains a positive integer *n*, indicating the number of maps to be processed. Each map follows. The first line describing a map contains a single positive integer, *k*, indicating the number of intersection points in the graph. These intersection points are automatically numbered 0 through *k-1*. The following *k* lines will contain *k* real numbers each, representing the distance between intersection *i* and intersection *j*, where i and j represent row and the column of the data (starting at 0.) It is guaranteed that the distance between an intersection and itself will be 0. The last line of each input case will store two integers which represent the starting and ending intersections, respectively for which you must determine the shortest path.

The Output:

At the beginning of each test case, output "`Map #`*p*", where *p* is the test case number (starting from 1) for the first line. For the second line, output the shortest distance with a statement in the following format:

```
The shortest distance between i and j is D.
```

where i and j are the intersection numbers for which the shortest path was requested, and D is that shortest distance, rounded to 2 decimal places.

The third line of output for the case should be a statement of the following format:

```
The shortest path from i to j is i->a->b->j.
```

where i and j are the desired starting and ending intersections and a and b are the intermediate stops. Note that there can be 0 or more intermediate stops; there certainly does not have to be two of them.

Leave a blank line after the output for each test case. Follow the format illustrated in Sample Output. Be sure to line up the output with spaces exactly as given in the Sample Output.

Sample Input:

```
2
3
0 1 2
3 0 5
2 4 0
2 1
3
0 2 2
1 0 3
4 1 0
0 1
```

Sample Output:

```
Map #1
The shortest distance between 2 and 1 is 3.00.
The shortest path from 2 to 1 is 2->0->1.

Map #2
The shortest distance between 0 and 1 is 2.00.
The shortest path from 0 to 1 is 0->1.
```