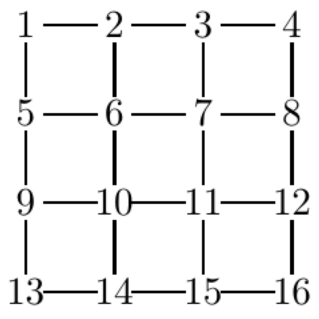
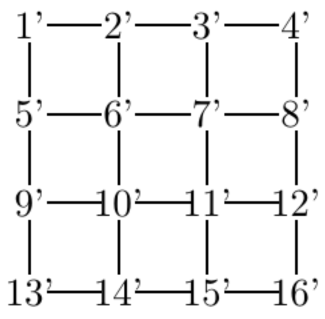
2586 Plugged In

The designers of the new NentindoBoxStation game system want to provide interactive input from many different sources. Using a special sensor-lined “electronic cocoon” users should be able to do things such as control simulated laser cannons by moving their eyebrows, accelerate/decelerate by wiggling their ears, steer in three-dimensional space by rotating their ankles ... the possibilities are endless.

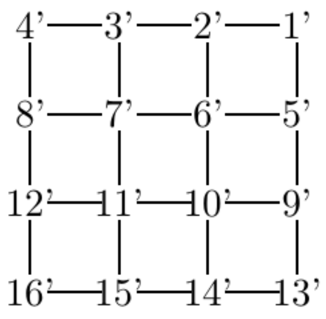
The connections between the sensors in the cocoon and the simulated actions in the computer are to be made using a special square plug with n^2 pins (the value of n has not yet been determined). Each pin can carry the output from one sensor, although for some applications not all pins will be active. The plug fits into a square socket containing n^2 holes that is attached to the various game inputs; again, for some games, not all input holes will be used. The socket can be flipped over and rotated to achieve different matchings between sensor pins and game inputs. Pins and socket holes are numbered consecutively in row major order (as shown below for the value $n = 4$).



Plug pins



Socket holes (front of plug)



Socket hole (back of plug)

Clearly pin 1 can be connected only to holes 1’, 4’, 13’, or 16’ (depending on how the plug is rotated with respect to the socket). Pin 2 can be connected only to holes 2’, 3’, 5’, 8’, 9’, 12’, 14’, or 15’ (if we consider all rotations and connections in both the back and front of the plug).

Most games require extra wiring to achieve connections because there is no way to match pins directly to their corresponding sockets (for instance, connecting pin 1 to hole 11’ in the figure). This wiring will be achieved with a special game-specific “wiring block” that will be placed between the plug and the socket. The lengths of these wires will depend on the orientation of the socket with respect to the plug. Given a list of connections that must be made, you are going to help the designers determine the *minimum average wire length* that is needed for the connections in the wiring block. Wires always run parallel to the grid lines, so the amount of wire between a pin in the plug and a hole in the socket is 1 plus the length of a shortest grid path between the nodes (the extra “1” is due to the thickness of the wiring block itself). Thus, one unit of wire is the minimum required (when a pin is positioned directly over the hole it is supposed to connect to).

For instance, if we are given the set (1,3’), (5,7’), (2,6’) for the plug and socket above, the average distance for this set of pairs is 2.6667 if we put the plug into the front of the socket without rotating the socket, but is only 2.3333 if we rotate the socket 180 degrees and then flip it horizontally, placing the plug in the back of the socket.

Input

Input will consist of a set of scenarios. Each scenario consists of a positive integer n, the side length of the plug and socket (less than or equal to 100) on a line by itself, followed by a positive integer m (less than or equal to n^2) on a line by itself, followed by m lines, each containing a pair of positive integers in the range $1, \ldots, n^2$. You may assume that no two pairs will have either a common first element or a common second element. The first integer represents a pin position in the plug, the second is a hole position in the socket. The final scenario is followed by '`0`' on a line by itself.

Output

For each scenario, output the scenario number (starting with 1), followed by the smallest average distance achievable between the m pin/socket pairs after rotations and reflections are considered (assuming an appropriate routing box is used), in a line of the form:

`Scenario` n`: smallest average =` avg

where avg is the average is rounded, and displayed, to four decimal places.

Separate lines of output by a single blank line.

Sample Input

```
4
3
1 3
5 7
2 6
2
3
1 4
2 2
4 1
0
```

Sample Output

```
Scenario 1: smallest average = 2.3333

Scenario 2: smallest average = 1.0000
```