Continental drift refers to the idea that once the earths continents were all in one piece and with time they drifted away from each other and arrived at their present positions.

In this problem, we are concerned with the reverse process — *Continental Coalesce*.

For the sake of brevity, lets assume the continents were initially merged together forming a square. With time, the square disintegrated into K different continents in the shape of rectangles and drifted away from its source.

You are provided with the current locations of the continents (i.e. rectangles). These locations were outlined with the help of satellites. Your first job is to figure out whether the data provided is a valid one. The data is valid if you can move the rectangles and form a square. In one move, you can pick any rectangle and push it one unit in one of the four directions (north, south, east or west). The continents can slide underneath one another during movement — meaning, more than one continent can occupy a particular position simultaneously. If it is possible to form a square then you are also required to find the minimum number of moves to do that. The final position of the square is not the primary concern here. Note that in the final position, two continents cannot overlap and the desired square cannot have any holes in it.

Input

The first line of input is an integer T ($T \leq 200$) that indicates the number of test cases. Each case consists of 20 lines containing 20 characters. Each character can either be a dot(`.`) representing an empty space or an '`x`' (ASCII value 120). Every '`x`' character will be part of some continent.

Notes:

- In the input grid, its guaranteed that, '`x`' characters from different continents will not be adjacent to each other. Any '`x`' character will be part of some rectangle
- Two cells are adjacent if they share a common edge.
- There will be exactly 25 '`x`' characters in the grid.
- The number of continents, K, will be at most 5.
- There is a blank line before the start of every case.
- The rectangles and the desired square are axis parallel.
- The earth here is flat. That is, the first and last row is NOT adjacent to each other. Same thing holds for the first and last column.

Output

For each case, output the case number first. If it's not possible to form a square by moving the rectangles around, output '`invalid data`', quotes for clarity. If it is possible, output the minimum number of moves required to complete the task.

Sample Input

```
2

....................
....................
....................
....................
....................
....................
..xxxxx.............
..xxxxx.............
....................
....................
..xx..........xxx...
....................
..xxxxx.............
..xxxxx.............
....................
....................
....................
....................
....................
....................

..................x..
....................
....................
....................
..xx................
..xx................
..xx................
..xx................
..xx................
..xx................
.......xxxxxx.......
.......xxxxxx.......
....................
....................
....................
....................
....................
....................
....................
....................
```

Sample Output

```
Case 1: 13
Case 2: invalid data
```