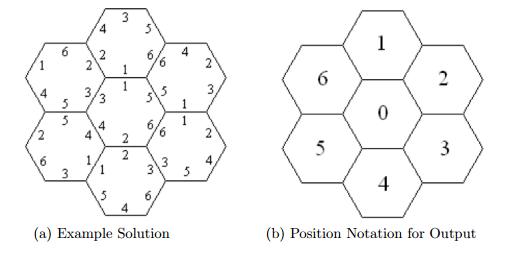
**PerplexaHexaFlexa**

PerplexaHexaFlexa, is a well known puzzle played by the inhabitants of the *Death Star* when Darth Vader isn’t looking. Darth Vader secretly knows the Death Star employees are playing the game behind his back, but is afraid to say anything because he isn’t very good at playing PerplexaHexaFlexa. In fact, he sucks at it. That’s where you come in. Lord Vader has hired an Intern(You!) from the Empire’s highest ranked Computer Science School in Cloud City to write a program to solve the puzzles for him, and then display the solution for him inside of his Helmet so know one knows.

The puzzle consists of 7 hexagonal pieces, each with the numbers 1 through 6 printed on the sides (Each Side representing a Star Wars movie name , so #1 = The Phantom Menace, #2 = Attack of the Clones, #3 = The Revenge of the Sith, #4 = Return of the Jedi, #5 = The Empire Strikes Back, and #6 = The Return of the Jedi). Each piece has a different arrangement of the numbers on its sides, and the object is to place the 7 pieces in the arrangement shown below such that the Movie names(Hint: The Numbers) on each shared edge of the arrangement are identical. Figure (a) is an example of one solution:



**Note:** You may notice that rotating any solution also gives another trivially identical solution. To avoid this redundancy, PerplexaHexaFlexa only deals with solutions which have a 1 on the uppermost edge of the central piece, as in the example above.

**Input:**

The first line of the input file will contain a single integer indicating the number of test cases. Each case will consist of a single line containing 42 integers. The first 6 represent the values on piece 0 listed in clockwise order; the second 6 represent the values on piece 1, and so on.

**Output:**

For each test case, output the case number (using the format shown below) followed by either the phrase No solution or by a solution specification. A solution specification lists the piece numbers in the order shown in the Position Notation of Figure (b). Thus if piece 3 is in the center, a 3 is printed first; if piece 0 is at the top, 0 is printed second, and so on. Each test case is guaranteed to have at most one solution.

**Sample Input:**

|  |
| --- |
| 2  3 5 6 1 2 4 5 1 2 3 6 4 2 3 5 4 1 6 3 1 5 6 2 4 5 4 1 3 6 2 4 2 3 1 5 6 3 6 1 2 4 5  6 3 4 1 2 5 6 4 3 2 5 1 6 5 3 2 4 1 5 4 6 3 2 1 2 5 6 1 4 3 4 6 3 5 2 1 1 3 5 2 6 4 |

**Sample Output:**

|  |
| --- |
| Case 1: 3 0 5 6 1 4 2  Case 2: No solution |