The 8-puzzle is a square tray in which eight square tiles are placed. The remaining ninth square is uncovered. Each tile has a number on it. A tile that is adjacent to the blank space can be slid into that space. A game consists of a starting state and a specified goal state. The starting state can be transformed into the goal state by sliding (moving) the tiles around. The 8-puzzle problem asks you to do the transformation in minimum number of moves.

2	8	3				1	2	3
1	6	4	=>			8		4
7		5				7	6	5
Start						Goal		

However, our current problem is a bit different. In this problem, given an initial state of the puzzle your are asked to discover a goal state which is the most distant (in terms of number of moves) of all the states reachable from the given state.

Input

The first line of the input file contains an integer representing the number of test cases to follow. A blank line follows this line.

Each test case consists of 3 lines of 3 integers each representing the initial state of the puzzle. The blank space is represented by a '0' (zero). A blank line follows each test case.

Output

For each test case first output the puzzle number. The next 3 lines will contain 3 integers each representing one of the most distant states reachable from the given state. The next line will contain the shortest sequence of moves that will transform the given state to that state. The move is actually the movement of the blank space represented by four directions: 'U' (Up), 'L' (Left), 'D' (Down) and 'R' (Right).

After each test case output an empty line.

Sample Input

```
1

2 6 4
1 3 7
0 5 8
```

Sample Output

```
Puzzle #1
8 1 5
7 3 6
4 0 2
UURDDRULLURRDLLDRRULULDDRUULDDR
```