

Two-Tiling

Problem

Your game company just ordered a lot of square game boards with 64 blank unit cells, intending to turn them into chessboards, but your boss has suddenly declared that chess is out of fashion. To make the best use of all of these boards, you have designed a new puzzle that uses *tiles*.

A tile is a contiguous set of unit cells, connected edge-to-edge, that can fit inside a 3x3 square of unit cells. For example, the following are examples of valid tiles (with each @ denoting a piece of the tile, and extra . characters for padding):

```
. . . @@@ @@@ .@@
. . . @@@ @.@ @.@
.@. @@@ @.. @@@
```

The following would NOT be valid tiles:

```
@@. @.@ .@@.
... .@. @@@@
.@@ @.@ .@@.
```

When the solver of your new puzzle places a tile on the board, its unit cells must exactly overlap some unit cells on the board that have not already been covered by other tiles. A tile is still considered the same type of tile even after being arbitrarily translated, rotated (by multiples of 90 degrees), and/or reflected, and the solver is allowed to do any of those things to a tile while placing it. For example, all of these are considered to be the same tile (and other variants of that tile are possible):

```
.@. ..@ @.. ... @@.
@@. .@@ @@. .@@ .@@
@.. .@. .@. @@. ...
```

To make your puzzle, you will color one or more of the cells on the board red. The solver will solve the puzzle by placing tiles on the board such that all red cells are covered up, but no other cell is covered up. To save on manufacturing costs, the solver receives only one type of tile, but they are given exactly enough copies of it to be able to cover all of the red cells.

Your job is to decide which of the board's cells to color red. Unfortunately, your boss is still deciding which of two particular tiles to use for the game. You are tired of waiting, so you have decided to try to color a set of cells such that the puzzle can be solved regardless of which of the tiles ends up being used.

Input

The first line of the input gives the number of test cases, **T**. **T** test cases follow; each consists of four lines. Each of the first three lines has three characters, then a single space, then three more characters. The fourth line is a blank line.

When looking at an entire case, the space characters separate a 3x3 grid on the left and a 3x3 grid on the right. Each grid represents a frame in which one of the two tiles is displayed. In each grid, each character is either @, representing a cell that is part of the tile, or ., representing a cell that is not part of the tile. Note that these . cells have nothing to do with the puzzle or the board,

and are just padding to make the shape of the tile clear. It is guaranteed that the two tiles are not the same, as described in the statement above.

Output

For each test case, output one line with `Case #x: y`, where `x` is the test case number (starting from 1), and `y` is `POSSIBLE` if there is a solution to the problem, and `IMPOSSIBLE` if there is not. Then, if there is a solution, output eight more lines of seventeen characters each, forming two 8x8 grids with one column of space characters in between. Each grid must use dot (.) characters to denote any blank cells, or characters from the following set of 64:

`! ? 0 1 2 3 4 5 6 7 8 9 a b c d e f g h i j k l m n o p q r s t u v w x y z A B C D E F G H I J K L M N O P Q R S T U V W X Y Z`

to denote the individual tiles used in a solution to the puzzle. Within each 8x8 grid, each non-dot character must denote a part of the same single tile, and different characters must denote different tiles. Each tile in the grid on the left must be the same as the tile on the left in input, up to rotations, translations and reflections. Each tile in the grid on the right must be the same as the tile on the right in input, up to rotations, translations and reflections. The set of cells that are not dots in both 8x8 grids must be the same, and must be nonempty.

If there are multiple valid solutions, you may output any one of them.

Limits for Test set 1 (Visible; the only test set)

T = 595. (Every possible test case, up to isomorphism, is included.)

Time limit: 30 seconds.

Memory limit: 1GB.

The cells in each tile in the input form a single contiguous group via their edge-to-edge connections.

The two tiles in the input are not the same, as described in the statement.

Sample

Sample Input

```
4
.@@ .@.
.@. .@.
.@@ @@.

@@@ @@@
@.@ @@@
@@@ @@@

.@. ...
@@. .@@
@.. ...

... ..@
... ..@
@.. ...
```

Sample Output

```
Case #1: POSSIBLE
....11.. ....11..
...221.. ...221..
...211.. ...321..
...22.... ...32...
.333.... .433....
4343.... 5444....
444.... 555....

.....
Case #2: IMPOSSIBLE
Case #3: POSSIBLE
.....
..T..I.. ..T..I..
.TT..II. .tT..Ii.
.T....I. .t....i.
.....
.LL..EE. .LL..EE.
..LLEE.. ..llee..
.....
Case #4: POSSIBLE
```

```
the.CODE AAB.CDDE
Jam.2018 FFB.CGGE
.....
World... HHIIJ...
.FiNALS. .KLLJMM.
.cup.... .KNN....
.....
TRIUMPH! OOPPQQRR
```

The sample output displays one set of answers to the sample cases. Other answers may be possible.

In Sample Case #2, there is no possible set of red cells that would make the puzzle solvable regardless of which of the two tiles is chosen.

In Sample Cases #3 and #4, note that the chosen set of red cells is not required to be contiguous. Also note that the dots in the input for a tile are not considered part of the tile, and have no significance in creating the puzzle. For example, the given answer would also be acceptable for the following input:

```
... ..
.@. .@.
... .@.
```

Moreover, that input is isomorphic with Sample Case #4, and would not appear in the same test set along with Sample Case #4.