

## Problem H: Chambers Ceramic Conundrum

The Chambers Construction Company has a major contract to deliver a tile floor on schedule for its largest customer (Moneybags to Spare Inc). Unfortunately, the clerk who ordered tiles won the lottery just after ordering the tiles for this room, and did not write down where to place each tile to make the room fit. Normally this would not be a problem, except the tiles that were ordered are not all squares. They are each made up of 4 square segments, but will take on all possible shapes shown here:

XXXX	XX	XX	XX	X	XXX	X
	XX	XX	XX	XXX	X	XXX

Given that the project is under an extremely tight schedule, it is not possible to reorder the tiles in a more standard manner. Instead when the 9 tiles come in, you will need to figure out how to place the tiles (or if there is no way to set the pieces correctly). The tiles in the box are ordered from A to I. The room that needs to be tiled is 6 segments on each side.

### Placing Tiles

A middle-manager at CCC has come up with an algorithm that they will give to the tiler to tile the room. The tiler has come to you to write a program to determine what pattern will be successful without having to try all of them with the physical tiles. The tiler will always start with the top left corner of the room. After placing the first tile, they will work their way from left to right and from top to bottom, placing the next tile such that it will fill the leftmost open space on the top line with an open space.

For instance, if the layout currently had the following tiles (A and B) placed:

```
AABBBZ
AA B
```

The next tile to place would be placed such that position Z is filled.

To make matters easiest, the tiler will always place the earliest tile in the box that could successfully fill that position. For instance, if either tile B or C could fill that position, the tiler will choose B. Furthermore, they will always place the tile as they are oriented above if possible. They will then attempt to rotate the tile 90 degrees clockwise and place it (possibly doing this 3 times).

NOTE: Remember the tiles may be rotated, but they may not be flipped.

A tile may not extend outside the 6x6 room, or overlap with another tile. A room is considered successfully tiled if the given set of tiles completely tile the room using the above algorithm.

The tiler will continue laying tiles according to this algorithm until finishing the room, or discovering that the room cannot be finished using the previous choices. If the room cannot be finished, the tiler will backtrack, considering the remaining rotations of the previous tile, and then the remaining tile. The tiler will continue to backtrack, one tile at a time, until all combinations have been tried or the room cannot be successfully tiled.

## Input

The first line of input will indicate how many data sets are included. ( $N$ )

The next  $N$  lines will each contain 9 numbers. These numbers indicate the shape of each tile. The first entry on the line will correspond to tile A, the second to tile B, ..., the 9th for tile I. Each number will reflect the layout of a tile as specified above (the left most layout is 1, the rightmost is 7).

## Output

For each data set, the first line of output should indicate the index of the data set, starting with 1. ("Data Set 1")

The next line of input will indicate the floor may or may not be tiled successfully. ("The floor may be tiled." or "The floor may not be tiled.").

The next 6 lines would then display the graph of floor illustrating the final layout. To make it easy to understand, each tile set should be marked A–I corresponding to the order they were in the input line. Each line will have exactly 6 characters indicating the tile segment, followed by a newline.

A blank line should appear after each data set.

The line stating "End of Output" should appear after the last data set.

## Example

Input:

```
2
2 2 2 2 2 2 2 2 3
1 1 2 1 1 2 1 1 2
```

Output:

```
Data Set 1
The floor may not be tiled.
```

```
Data Set 2
The floor may be tiled.
AAAABD
CCEGBD
CCEGBD
FFEGBD
FFEGII
HHHHII
```

```
End of Output
```