Problem G. Play the game SET

Input file:	`standard input`
Output file:	`standard output`
Time limit:	15 seconds
Memory limit:	256 megabytes

SET is a real-time card game designed by Marsha Falco in 1974 and published by Set Enterprises in 1991. The deck consists of 81 unique cards that vary in four features across three possibilities for each kind of feature: number of shapes (one, two, or three), shape (diamond, squiggle, oval), shading (solid, striped, or open), and color (red, green, or purple). Each possible combination of features (e.g. a card with [three][striped][green][diamonds]) appears as a card precisely once in the deck.

In the game, certain combinations of three cards are said to make up a **set**. For each one of the four categories of features — color, number, shape, and shading — the three cards must display that feature as a) either all the same, or b) all different. Put another way: For each feature the three cards must avoid having two cards showing one version of the feature and the remaining card showing a different version.

For example,

[three][solid][red][diamonds]

[two][solid][green][squiggles]

[one][solid][purple][oval]

form a **set**, because the shadings of the three cards are all the same, while the numbers, the colors, and the shapes among the three cards are all different.

You're given **N** unique cards, you need to form them into sets. You need to output the maximum number of sets with the given cards, and each card is used at most once.

Input

The first line of the input gives the numbers of test cases, **T**. **T** test cases follow. Each test case consists of one line with one integer **N** as described above. The i-th of the next **N** lines describes the i-th card. It contains four categories features, in the format of "[*][*][*][*]"

There are no identical cards.

$1 \leq \mathbf{T} \leq 100$.

$1 \leq \mathbf{N} \leq 38$.

There are at most 10 test cases have $\mathbf{N} > 35$.

Output

For each test case, the first line print "Case #x: y", where x is the test case number (starting from 1) and y is the maximum number of sets. Then print y triples $id1$, $id2$, $id3$, defining the indexes of cards (starting from 1) that can make up a **set**. If there are several solutions, output any of them.

Example

standard input	standard output
1 7 [one][diamond][solid][red] [two][diamond][solid][red] [three][diamond][solid][red] [one][diamond][solid][green] [one][diamond][solid][purple] [two][diamond][solid][purple] [purple][three][diamond][solid]	Case #1: 2 1 2 3 5 6 7