D – Glimmr in Distress

In a special compression technique used by a photo sharing site Glimmr to compress its database of images, a simple preprocessing is used. As you may know, any digital image is comprised of several matrices, one for each channel, of binary values. Each of these matrices can further be broken up into a couple of Boolean matrices associated with binary places (2^0 2^1 2^2…). In other words, each digit of a binary pixel value appears in the corresponding coordinates of a B&W (black-and-white) image pertinent to the digit's place-value. Thus, we can represent a color image with a collection of same-size B&W images. The preprocessing phase is carried out on each B&W image which without loss of generality we assume to be of size $2^n \times 2^n$. As a result, the image is converted to a string of characters '0', '1' and '2'. More specifically, the string representation of an image of uniform 1s is a single-character string "1" and that of uniform 0s, the string "0". For non-uniform values, the image is split into four sub-images of size $2^{n-1} \times 2^{n-1}$, as seen below. The string translation in this case starts with the character '2' followed by the string codes for sub-images in order from left to right, top to bottom, that is $S_1S_2S_3S_4$.

I_1	I_2
I_3	I_4

For example, the string representing the following 2×2 image is "21001".

1	0
0	1

These strings are further processed to achieve a good level of compression and then stored. Unfortunately, a recent act of sabotage (probably with an EMP micro bomb) by a rival company has left their database damaged with garbled values. As a result, when one wants to decompress and reconstruct images they run into strings that may have some characters altered and some others lost. Therefore, Glimmr is pursuing a disaster recovery plan. To start with, they intend to identify damaged strings and for those seemingly unharmed, find their minimum consistent image size, since size information is also missing.

You are called in to help them with this task.

Input

The first line of input contains an integer T≤1,000 denoting the number of test-cases. Each test-case is represented by a single line containing a string of size of at most 2,500 of characters 0, 1 and 2.

Output

For each test-case, if the string is not a valid representation of a B&W image output "Not Possible" with no quotes. And if it is valid, output the minimum possible size of the image in the format 2^n*2^n as seen in the sample output.

Sample Input	Sample Output
4 2022111011111 2112002000001 20102102101010 1	2^3*2^3 Not Possible Not Possible 2^0*2^0