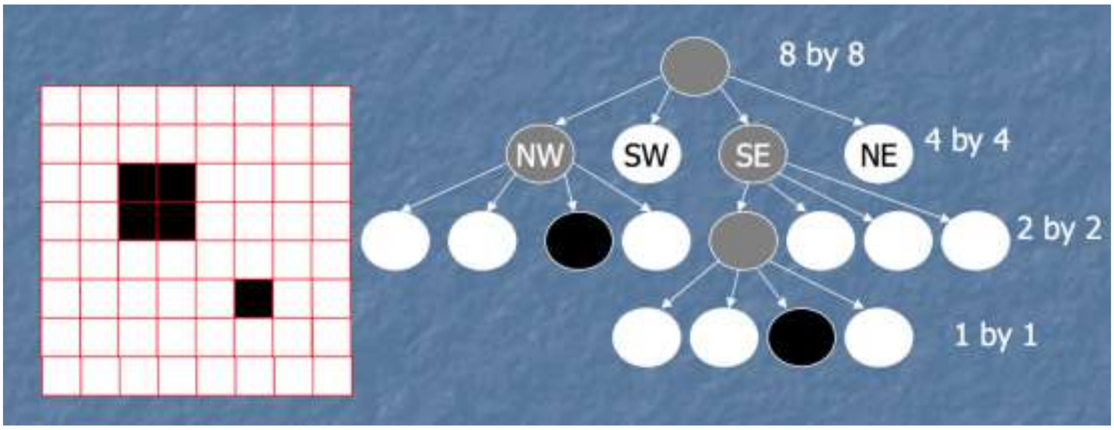
Simple Quadtree

As we all know, images consist of many tiny pixels. Have you ever considered how we can efficiently store the picture in computers? There is a simple data structure called "Quadtree", which has appeared in the first lecture notes.

A quadtree is a representation format used to encode images. The fundamental idea behind the quadtree is that any image can be split into four quadrants. Each quadrant may again be split into four sub quadrants, etc. In the quadtree, the image is represented by a parent node, while the four quadrants are represented by four child nodes, in a predetermined order. Of course, if the whole image only contains a single color, it definitely can be represented by a quadtree consisting of a single node. In general, a quadrant needs only to be subdivided if it consists of pixels of different colors. As a result, the quadtree need not be of uniform depth.

In this question, given a picture containing only black and white pixels, please find out the number of nodes (including the root) in the corresponding quadtree of the picture. It is not necessary to implement the data structure of quadtree.



Input

The input contains multiple cases. Each test case begins with one integer k, indicating the size $(\mathrm{n} \times \mathrm{n})$ of image, where $n = 2^k$ and $0 \leq \mathrm{k} \leq 10$.

The following n lines give the information of the pixels, where the i-th line contains a binary string (containing '0'(white) and '1'(black)) of n characters, indicating the pixel information in the i-th row.

Output

For each test case print the number of nodes in the corresponding quadtree in a separate line.

Sample input	**Sample output**
2 0011 0001 1111 0111 2 0000 0000 0000 0000 3 11111101 01101011 10000000 10010111 00101111 01011101 11110100 11000010 1 11 10	13 1 77 5