

## J2 – Blur

You have a black and white image that is  $w$  pixels wide and  $h$  pixels high. You decide to represent this image with one number per pixel: black is 0, and white is 1. Your friend asks you to blur the image, resulting in various shades of gray. The way you decide to blur the image is as follows: You create a new image that is the same size as the old one, and each pixel in the new image has a value equal to the average of the 9 pixels in the  $3 \times 3$  square centered at the corresponding old pixel. When doing this average, wrap around the edges, so the left neighbor of a leftmost pixel is in the rightmost column of the same row, and the top neighbor of an uppermost pixel is on the bottom in the same column. This way, the  $3 \times 3$  square always gives you exactly 9 pixels to average together. If you want to make the image blurrier, you can take the blurred image and blur it again using the exact same process.

Given an input image and a fixed number of times to blur it, how many distinct shades of gray does the final image have, if all the arithmetic is performed exactly?

*Warning: Floating point numbers can be finicky; you might be surprised to learn, for example, that  $2/9 + 5/9$  may not equal  $3/9 + 4/9$  if you represent the fractions with floating point numbers! Can you figure out how to solve this problem without using floating point arithmetic?*

### Input

The first line of input contains three space-separated integers  $w$ ,  $h$ , and  $b$  ( $3 \leq w, h \leq 100, 0 \leq b \leq 9$ ), denoting the width and height of the image, and the number of times to blur the image, respectively. The following  $h$  lines of  $w$  space-separated integers describe the original image, with each integer being either 0 or 1, corresponding to the color of the pixel.

### Output

Output, on a single line, a single integer equal to the number of distinct shades of gray in the final image.

### Input and output samples

Input :	Output :
5 4 1	3
0 0 1 1 0	
0 0 1 1 0	
0 0 1 1 0	
0 0 1 1 0	
0 0 1 1 0	

  

Input :	Output :
3 3 2	1
1 0 0	
0 1 0	
0 1 0	