Problem H
Where's Wally

Input: H.in
Time Limit: 90 seconds

Do you know the famous series of children's books named "Where's Wally"? Each of the books contains a variety of pictures of hundreds of people. Readers are challenged to find a person called Wally in the crowd.

We can consider "Where's Wally" as a kind of pattern matching of two-dimensional graphical images. Wally's figure is to be looked for in the picture. It would be interesting to write a computer program to solve "Where's Wally", but this is not an easy task since Wally in the pictures may be slightly different in his appearances. We give up the idea, and make the problem much easier to solve. You are requested to solve an easier version of the graphical pattern matching problem.

An image and a pattern are given. Both are rectangular matrices of bits (in fact, the pattern is always square-shaped). 0 means white, and 1 black. The problem here is to count the number of occurrences of the pattern in the image, i.e. the number of squares in the image exactly matching the pattern. Patterns appearing rotated by any multiples of 90 degrees and/or turned over forming a mirror image should also be taken into account.

Input

The input is a sequence of datasets each in the following format.

> w h p
> image data
> pattern data

The first line of a dataset consists of three positive integers w, h and p. w is the width of the image and h is the height of the image. Both are counted in numbers of bits. p is the width and height of the pattern. The pattern is always square-shaped. You may assume $1 \leq w \leq 1000$, $1 \leq h \leq 1000$, and $1 \leq p \leq 100$.

The following h lines give the image. Each line consists of $\lceil w/6 \rceil$ (which is equal to $\lfloor (w+5)/6 \rfloor$) characters, and corresponds to a horizontal line of the image. Each of these characters represents six bits on the image line, from left to right, in a variant of the BASE64 encoding format. The encoding rule is given in the following table. The most significant bit of the value in the table corresponds to the leftmost bit in the image. The last character may also represent a few bits beyond the width of the image; these bits should be ignored.

character	value (six bits)
A–Z	0–25
a–z	26–51
0–9	52–61
+	62
/	63

The last p lines give the pattern. Each line consists of $\lceil p/6 \rceil$ characters, and is encoded in the same way as the image.

A line containing three zeros indicates the end of the input. The total size of the input does not exceed two megabytes.

Output

For each dataset in the input, one line containing the number of matched squares in the image should be output. An output line should not contain extra characters.

Two or more matching squares may be mutually overlapping. In such a case, they are counted separately. On the other hand, a single square is never counted twice or more, even if it matches both the original pattern and its rotation, for example.

Sample Input

```
48 3 3
gAY4I4wA
gIIgIIgg
w4IAYAg4
g
g
w
153 3 3
kkkkkkkkkkkkkkkkkkkkkkkkk
SSSSSSSSSSSSSSSSSSSSSSSSQ
JJJJJJJJJJJJJJJJJJJJJJJJI
g
Q
I
1 1 2
A
A
A
384 3 2
ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789+/
BCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789+/A
CDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789+/AB
A
A
0 0 0
```

Output for the Sample Input

```
8
51
0
98
```