1114 – Easily Readable

As you probably know, the human information processor is a wonderful text recognizer that can handle even sentences that are garbled like the following:

The ACM Itrenntaoial Clloegaite Porgarmmnig Cnotset (IPCC) porvdies clolgee stuetnds wtih ooppriuntetiis to itnrecat wtih sutednts form ohetr uinevsrtieis.

People have claimed that understanding these sentences works in general when using the following rule: The first and last letters of each word remain unmodified and all the characters in the middle can be reordered freely. Since you are an ACM programmer, you immediately set on to write the following program: Given a sentence and a dictionary of words, how many different sentences can you find that could potentially be mapped to the same encoding?

Input

Input starts with an integer **T (≤ 20)**, denoting the number of test cases.

Each case starts with a line containing the number **n (0 ≤ n ≤ 10000)** of words in the dictionary, which are printed on the following **n** lines. After this, there is a line containing the number **m (0 ≤ m ≤ 10000)** of sentences that should be tested with the preceding dictionary and then **m** lines containing those sentences. The sentences consist of letters from **a** to **z**, **A** to **Z** and spaces only and have a maximal length of **10000** characters. For each word in the dictionary a limitation of **100** characters can be assumed. The words are case sensitive. In any case, total number of characters in the sentences will be at most 10^5. And total characters in the dictionary will be at most 10^5.

Output

For each case, print the case number first. Then for each sentence, output the number of sentences that can be formed on an individual line. Result fits into 32 bit signed integer.

Sample Input	Output for Sample Input
1 8 baggers beggars in the blowed bowled barn bran 1 beggars bowled in the barn	Case 1: 8

Note

Dataset is huge, use faster I/O methods.