I - String Shuffle

Input: `shuffle.in`
Output: `standard output`

Given three strings consisting of just lowercase letters, count the number of ways that the third string can be constructed by combining two *subsequences* from the first two strings.

One derives a *subsequence* of the string by deleting zero or more characters from a string. For example, "", "a", "b", "c", "ab", "ac", "bc", and "abc" are all the subsequence strings of "abc". (Note that the empty string, "", is a subsequence of *any* string.)

The two subsequences are combined to make a third string by *shuffling* them together. That is, the relative order of the letters from the subsequence cannot be changed in the target string; but the two subsequences can be interleaved arbitrarily. For example, consider the two subsequences "abc" and "de". By combining them, one can get the following strings: "abcde", "abdce", "abdec", "adbce", "adbec", "adebc", "dabce", "dabec", "daebc", and "deabc".

Input

The first line of the input contains a single integer t that indicates the number of test cases. Each test case contains 3 strings, each containing only lowercase characters. The length of each string is between 1 and 60, inclusive.

Output

For each test case, output a line with a single integer that denotes the number of ways that one can construct the third string from the first two strings as described above.

Sample Input

```
3
abc abc abc
aa aa aa
abbcd bccde abcde
```

Sample Output

```
8
10
18
```