1416 – Superb Sequence

There were three friends (Alice, Bob and Carol) who regularly went to expeditions and discovered new mountain peaks. They often proposed different names and it was a problem to decide which name they would choose for the newly discovered peaks. Alice and Bob both said that the name of the peak must be a super sequence of their proposed names **A** and **B**, i.e. **A** and **B** should be **subsequences** of the name of the peak. Carol said that the name of the peak must be a **subsequence** of her proposed name **C**. As they don't like long names, they want to know the number of distinct shortest names which satisfy their needs.

So, given three strings **A, B** and **C**, you have to find the number of distinct shortest **common super sequences** of **A** and **B** who are also a **subsequence** of **C**. Moreover, you need to find the lexicographically earliest such sequence. Two sequences are distinct if they differ in at least one position. A **subsequence** is a sequence obtained by deleting zero or more characters from a string. A **super-sequence** is a sequence obtained by inserting zero or more characters in one or more positions of the string.

For example, say, **A = "cdfa"**, **B = "dga"** and **C = "bcdfgaga"**. Then there are two shortest common super sequences of **A** and **B**: **"cdfga"** and **"cdgfa"**, but **"cdgfa"** is not a subsequence of **C**. So the only possible name for the peak is **"cdfga"**.

Input

Input starts with an integer **T (≤ 100)**, denoting the number of test cases.

Each case contains three lines. First line contains a string denoting **A**, second line contains **B** and third line contains **C**. Assume that the strings are non-empty and length of **A** and **B** will not be more than **100** and length of **C** will not be more than **300**.

Output

For each case, print the case number and the number of distinct possible shortest names for the peak modulo **1000 000 007**. And second line should contain the lexicographically earliest name. If no solution is found then print **"NOT FOUND"** in second line.

Sample Input	Output for Sample Input
2 cdfa dga bcdfgaga abc defm abcdfghm	Case 1: 1 cdfga Case 2: 0 NOT FOUND