**Anagram**

We say that two English words is a pair of anagrams if the two words consists of the same set of alphabets (up to the number of repetitions of a same letter), possibly different in the ordering of the letters. For example, “more” and “rome” is a pair of anagrams.

Let’s define a stronger form of anagrams (called *strong anagrams*). In this definition, when you rearrange the letters of the first word to match the second word, the letters are allowed to move only within a distance of D. For example, if D=2, “more” and “rome” is a pair of strong anagrams. But if D=1, they are not.

Write a program, given the value of D and two English words, that determines if the given words form a pair of strong anagrams.

[Input]

In the first line of the input file is given the number of test cases T. (T ≤ 60) Each test case consists of 3 lines. The first line contains D and D is an integer between 0 and 100,000, inclusive. Each of the next 2 lines contains an English word consisting of lower-case letters. The length of a word is between 1 and 100,000, inclusive. The two words may not be a pair of anagrams, even differing in lengths.

The input is given in 3 sets as follows:

* Set 1: The length of a word is at most 10.
* Set 2: The length of a word is at most 1,000.
* Set 3: The length of a word is at most 100,000.

[Output]

For each test case, print an O (upper case letter) if the given words is a pair of strong anagrams, an X if not.

[Input/Output Example]

Input

|  |
| --- |
| 3  2  more  rome  1  more  rome  3  anagram  magran |

Output

|  |
| --- |
| O  X  X |