

Pattern Search

Input file: **standard input**
Output file: **standard output**
Time limit: 3 seconds
Memory limit: 512 megabytes

Bajtek is attending the course Introduction to Algorithms. Today, he learned efficient algorithms to solve the problem of pattern searching within a text. He already implemented an efficient program that reads from input the text s and pattern t , both made up of lowercase English letters, and determines the number of occurrences of pattern t as a substring of text s .

To test the correctness of his program, Bajtek generated a sample text s and pattern t . The program gave the correct answer, but Bajtek did not like that correct answer... He very much wanted to modify his test so that the number of pattern occurrences was as high as possible. Unfortunately, Bajtek's text editor struggles with editing large files. It only allows for arbitrary changes in the order of letters in the text s and – separately – for arbitrary changes in the order of letters in the pattern t .

Can you help Bajtek and determine the maximum number of occurrences of the pattern t in the text s if Bajtek makes an optimal edit to his test?

Input

The first line of input consists of one integer z ($1 \leq z \leq 100\,000$) – the number of independent test sets. The description of each test set is in a separate line and consists of two words s and t , made up of lowercase English letters.

You can assume that $1 \leq |t| \leq |s| \leq 2\,000\,000$. Furthermore, the total length of words in the input file will not exceed 4 000 000 characters.

Output

The output should have z lines; the i -th of them should contain one integer – the maximum number of occurrences of the pattern in the text after editing Bajtek's i -th test set.

Example

standard input	standard output
2	2
bajkaaall aal	1
abca cba	

Note

Explanation of the first example: Bajtek can change the order of letters in the text s to **balalajka**, and in the pattern t – to **ala**. After editing, the pattern appears twice in the text.