HLCS

SEASON 8 – SIXTH ROUND

The top programmer Tanya has recently learned about **LCS** (**L**ongest **C**ommon **S**ubstring) problem. As this problem was too easy for her, she came up with harder version of it.

In this new version, you are given two strings **S** and **T**, and **Q** queries. For every query you are given a pair of integers **(L[i]**, **R[i])**, that define the ***substring of consecutive characters*** of **S** between these two positions – **S(L[i]; R[i])**. Both border characters are included in this substring.

For every query you must find the value of **LCS(S(L[i]; R[i]) , T)**, i.e. the length of the longest common subsequence of **consecutive** characters for the strings **S(L[i]; R[i])** and **T**. If there is no common subsequence between the two strings, the answer to the query is 0.

Unfortunately, Tanya has troubles with this new problem. Help her, by writing a program **HLCS**, which solves this harder version of the **LCS** problem.

Input

The first two lines of the input file hlcs.in contain the strings **S** and **T**. The third line contains **Q** – the number of queries.

The next **Q** lines contain the borders of the queries - **L[i]** and **R[i]**.

Output

The output file hlcs.out must containt **Q** lines with one integer on each – the answers to the queries.

Constraints:

1 ≤ |S|, |T|, Q ≤ 300 000

1 ≤ L[i] ≤ R[i] ≤ |S|

Time limit: 1.5 sec

Memory limit: 256 MB

Example tests:

Input (hlcs.in)	Output (hlcs.out)
abcdaaaaa daaaab 3 1 2 1 9 3 3	2 5 0
iwqeiowuqeoiuwqoieuoiwqsahdlkhjslk sadkjkiwqeiqoieusadwqe 4 1 10 1 20 4 10 1 30	5 5 2 5