Benjamin's granddaughter Brenda has a blog where she posts articles about school, friends and other life issues. Intrigued by her opinions, Benjamin tried to read it, but very soon he realized it was too hard to read because of Brenda's writing quirks.

Brenda writes without spaces or punctuation marks, and moreover, she uses lower and uppercase letters in a liberal and strange way. For example, one of her posts is "`PrOgRAMmINgiSgrEAt`". Benjamin has trouble noticing the words "`programming`", "`is`" and "`great`" when they are written in this way.

To improve his understanding Benjamin decided to do the following: he will first choose a particular string T and a blog post he is interested in; then he will select a contiguous substring of the post and search for T within the substring, in a case-insensitive way; for each occurrence of T within the substring, he will calculate the number of case mismatches, and finally he will obtain the maximum among all these values. For example, if Benjamin chooses "`GR`" as T and then selects the substring "`PrOgRAM`", he would find a single occurrence "`gR`" for which the number of case mismatches is 1. For the same substring, if "`r`" was chosen as T, he would have found two occurrences, "`r`" with 0 mismatches and "`R`" with 1 mismatch, so the maximum number of mismatches would be 1.

To complicate things further, Brenda included in the blog a script that, after operating with a substring selection, flips the case of all the selected letters. This means that after selecting "`PrOgRAM`" and proceeding as explained above, the sample post would read "`pRoGrammINgiSgrEAt`". If Benjamin selects "`ammINgi`" as a second substring, after calculating his result the post would be left as "`pRoGrAMMinGISgrEAt`", accumulating both flips.

You will be given the string T and the original text of the blog post chosen by Benjamin. You will also be given a list of substring selections Benjamin made, in the order he made them. You need to calculate, for each selection, the maximum number of case mismatches of the occurrences of T in the selected part, considering all the case flips made by previous selections. Notice that the flipping of the case occurs after calculating the result for each selection.

Input

The input file contains several test cases, each of them as described below.

The first line contains an integer N ($1 \leq N \leq 10^5$) and a non-empty string T of at most 5 letters, representing respectively the number of substring selections and the string to search for. The second line contains a non-empty string P of at most 10^5 letters, indicating the original text of the blog post. Positions of the post are numbered with consecutive integers from left to right, being 1 the leftmost position and $|P|$ the rightmost position. Each of the next N lines describes a substring selection with two integers L and R ($1 \leq L \leq R \leq |P|$) indicating that the substring starts at position L and ends at position R, inclusive.

Output

For each test case, output N lines, each of them containing an integer. In the i-th line write the maximum number of case mismatches of the occurrences of T in the i-th substring selection, considering all the case flips made by previous selections; if no such occurrence exists write the value '`-1`'.

Sample Input

```
3 gR
PrOgRAMmINgiSgrEAt
1 7
4 18
6 14
9 abCAb
aBcAbCAbaBCAb
1 13
1 13
4 8
5 11
3 11
4 10
1 13
8 8
1 13
```

Sample Output

```
0
2
-1
2
4
1
-1
0
5
2
-1
4
```