D. DZY Loves Strings

time limit per test: 3 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

DZY loves strings, and he enjoys collecting them.

In China, many people like to use strings containing their names' initials, for example: `xyz`, `jcvb`, `dzy`, `dyh`.

Once DZY found a lucky string s. A lot of pairs of good friends came to DZY when they heard about the news. The first member of the i-th pair has name a_i, the second one has name b_i. Each pair wondered if there is a substring of the lucky string containing both of their names. If so, they want to find the one with minimum length, which can give them good luck and make their friendship last forever.

Please help DZY for each pair find the minimum length of the substring of s that contains both a_i and b_i, or point out that such substring doesn't exist.

A substring of s is a string $s_l s_{l+1} \ldots s_r$ for some integers l, r $(1 \le l \le r \le |s|)$. The length of such the substring is $(r - l + 1)$.

A string p contains some another string q if there is a substring of p equal to q.

Input

The first line contains a string s $(1 \le |s| \le 50000)$.

The second line contains a non-negative integer q $(0 \le q \le 100000)$ — the number of pairs. Each of the next q lines describes a pair, the line contains two space-separated strings a_i and b_i $(1 \le |a_i|, |b_i| \le 4)$.

It is guaranteed that all the strings only consist of lowercase English letters.

Output

For each pair, print a line containing a single integer — the minimum length of the required substring. If there is no such substring, output `-1`.

Examples

input
xudyhduxyz 3 xyz xyz dyh xyz dzy xyz
output
3 8 -1

input
abcabd 3 a c ab abc ab d
output

```
2
3
3
```

input

```
baabcabaaa
2
abca baa
aa aba
```

output

```
6
4
```

Note

The shortest substrings in the first sample are: `xyz`, `dyhduxyz`.

The shortest substrings in the second sample are: `ca`, `abc` and `abd`.

The shortest substrings in the third sample are: `baabca` and `abaa`.