

Frankenstein's Banner

time limit per test: 4 seconds (10 seconds for Python)

memory limit per test: 1 gigabyte

input: standard input

output: standard output

points: 100 · problem code: archerybannerstrips · author: Parsa

The MAPS Academy team is preparing the captions for an Olympic archery stream. The full chant for the day is one long lowercase string S , but the director keeps asking for short slices of it between ends, so the team has to assemble them from prepared banner strips.

The team has n reusable banner templates, written as strings $t_1, t_2, \dots, t_n$. There are infinitely many copies of every template, and using one copy has cost 1.

Before using a copy, the team may cut off any suffix, possibly the empty suffix. Equivalently, one copy may contribute any prefix of its template. The chosen prefixes may then be concatenated in any order.

For each request $[l_i, r_i)$, determine the minimum number of template copies needed to form exactly the substring $S[l_i, r_i)$. The indices are zero-based, and the right endpoint is not included. If the substring cannot be formed, output

-1.

Input

The first line contains two integers n ($1 \leq n \leq 10^4$) and m ($1 \leq m \leq 3 \times 10^5$): the number of banner templates and the number of requests.

The next n lines contain the nonempty strings t_i . All strings consist only of lowercase English letters. The total length of all strings t_i is at most 5×10^5 .

The next line contains the string S , consisting only of lowercase English letters ($1 \leq |S| \leq 3 \times 10^5$).

Each of the next m lines contains two integers l_i and r_i ($0 \leq l_i < r_i \leq |S|$), describing one requested substring.

Output

Print m lines. The i -th line must contain the answer to query i , or -1 if it is impossible.

Example 1

Input

```
3 8
aim
arrow
gold
aimarrowgoldaimx
0 3
3 6
0 8
0 15
8 15
1 3
15 16
0 16
```

Output

```
1
1
2
4
2
-1
-1
-1
```