

A substring is defined as a contiguous sequence of one or more characters in the string. More information on substrings can be found [here](#).

You are given  $n$  strings  $w[1], w[2], \dots, w[n]$ . Let  $S[i]$  denote the set of all unique substrings of the string  $w[i]$ . Let  $S = \{S[1] \cup S[2] \cup \dots S[n]\}$ , that is,  $S$  is a set of strings that is the union of all substrings in all sets  $S[1], S[2], \dots S[n]$ . There will be many queries. For each query you will be given an integer ' $k$ '. Your task is to find the  $k^{\text{th}}$  element of the **1-indexed** lexicographically ordered set of substrings in the set  $S$ . If there is no element  $k$ , return **INVALID**.

For example, your strings are  $w = [abc, cde]$ . All of the substrings are  $S[1] = \{a, b, c, ab, bc, abc\}$  and  $S[2] = \{c, d, e, cd, de, cde\}$ . Combine the two sets and sort them to get  $S = \{a, ab, abc, b, bc, c, cd, cde, d, de, e\}$ . So, for instance if  $k = 1$ , we return 'a'. If  $k = 5$ , we return 'bc'. If  $k = 20$  though, there is not an  $S[20]$  so we return **INVALID**.

### Function Description

Complete the *findStrings* function in the editor below. It should return array of strings.

*findStrings* has the following parameter(s):

- $w$ : an array of strings
- $queries$ : an array of integers

### Input Format

The first line contains an integer  $n$ , the number of strings in the array  $w$ .

Each of the next  $n$  lines consists of a string  $w[i]$ .

The next line contains an integer  $q$ , the number of queries.

Each of the next  $q$  lines consists of a single integer  $k$ .

### Constraints

$$1 \leq n \leq 50$$

$$1 \leq |w[i]| \leq 2000$$

$$1 \leq q \leq 500$$

$$1 \leq k \leq 10^9$$

Each character of  $w[i] \in \text{ascii}[a - z]$

### Output Format

Return an array of  $q$  strings where the  $i^{\text{th}}$  string is the answer to the  $i^{\text{th}}$  query. If a  $k$  is invalid, return "INVALID" for that case.

### Sample Input

```
2
aab
aac
3
3
8
23
```

### Sample Output

```
aab
c
INVALID
```

### Explanation

For the sample test case, we have 2 strings "aab" and "aac".

$S_1 = \{ "a", "aa", "aab", "ab", "b" \}$ . These are the 5 unique substrings of "aab".

$S_2 = \{ "a", "aa", "aac", "ac", "c" \}$ . These are the 5 unique substrings of "aac".

Now,  $S = \{ S_1 \cup S_2 \} = \{ "a", "aa", "aab", "aac", "ab", "ac", "b", "c" \}$ . Totally, 8 unique strings are present in the set  $S$ .

The lexicographically 3rd smallest string in  $S$  is "aab" and the lexicographically 8th smallest string in  $S$  is "c". Since there are only 8 distinct substrings, the answer to the last query is "INVALID".