**Problem Statement** **Suggest Edit**

**Ninja has given you a list of WORDS and a PATTERN string. Your task is to find all such words in the list which match the given pattern. A valid match is found if and only if every character in the pattern is uniquely mapped to a character in a word.**

**Example:**

```
Let the list of words be {cod, zcz} and the pattern be “nin”.
For each word in the list, we will check whether the word matches the pattern or
not:

For the word “cod”:
Letter ‘n’ in the pattern maps to letter ‘c’ in the word.
Letter ‘i’ in the pattern maps to letter ‘o’ in the word.
Letter ‘n’ in the pattern maps to letter ‘d’ in the word.

As the same letter ‘n’ in the pattern, maps to two different letters ‘c’ and ‘d’
in the word. Hence, “cod” is not a valid match.

For the word “zcz”:
Letter ‘n’ in the pattern maps to letter ‘z’ in the word.
Letter ‘i’ in the pattern maps to letter ‘c’ in the word.
Letter ‘n’ in the pattern maps to letter ‘z’ in the word.

As every letter in the pattern maps uniquely to a corresponding letter in the
word. Hence “zcz” is a valid match.
```

**Input Format:**

```
The very first line of input contains an integer ‘T’ denoting the number of test
cases.

The first line of every test case contains an integer ‘N’ denoting the number of
words present in the list.

The second line of every test case contains ‘N’ space-separated strings denoting
the words in the list.

The third line of every test case contains the pattern string.
```

**Output Format:**

```
For each test case, print space-separated words which are a valid match with the
given pattern.
```

**Note:**

Print the words in the same order in which they occur in the list.

You do not need to print anything, it has already been taken care of. Just implement the given function.

**Constraints:**

```
1 <= T <= 10
1 <= N <= 10^2
1 <= Length of pattern, Length of each word <= 10^3

Time Limit: 1 sec
```

**Sample Input 1:**

```
2
2
cdd pcm
foo
3
abcd km qst
pqr
```

**Sample Output 1:**

```
cdd
qst
```

**Explanation 1:**

For the first test case, the list of words is {cdd, pcm} and the pattern is “foo”.

For the word “cdd”:
The letters ‘f’, ‘o’, ‘o’ of the pattern, map to letters ‘c’, ‘d’, ‘d’ of the word respectively. As, every letter in the pattern maps uniquely to a corresponding letter in the word. Hence, it is a valid match.

For the word “pcm”:
The letters ‘f’, ‘o’, ‘o’ of the pattern map to letters ‘p’, ‘c’, ‘m’ of the word respectively. As the same letter ‘o’, in the pattern, maps to two different letters ‘c’ and ‘m’ in the word. Hence, “pcm” is not a valid match.

For the second test case, the list of words is {abcd, km, qst} and the pattern is “pqr”.

For the word “abcd”:
The letters ‘p’, ‘q’, ‘r’ of the pattern map to letters ‘a’, ‘b’, ‘c’ of the word respectively. But, there is no character in the pattern which maps to the letter ‘d’ in the word. Hence, it is not a valid match.

For the word “km”:
The letters ‘p’, ‘q’ of the pattern, map to letters ‘k’, ‘m’ of the word respectively. But. there is no character in the word which maps to the letter ‘r’ in the pattern. Hence, it is not a valid match.

For the word “qst”:
The letters ‘p’, ‘q’, ‘r’ of the pattern map to letters ‘q’, ‘s’, ‘t’ of the word respectively. As every letter in the pattern maps uniquely to a corresponding letter in the word. Hence, it is a valid match.

**Sample Input 2:**

```
2
5
aaaa abcd code toma zedi
pqrs
6
adff coding ejqq fstt ggnn ninja
lmnn
```

**Sample Output 2:**

```
abcd code toma zedi
adff ejqq fstt
```

**Sample Input 3:**

```
2
3
#h#@# AmAka &t&y&
%R%s%
1
A
B
```

**Sample Output 3:**

```
#h#@# &t&y&
A
```