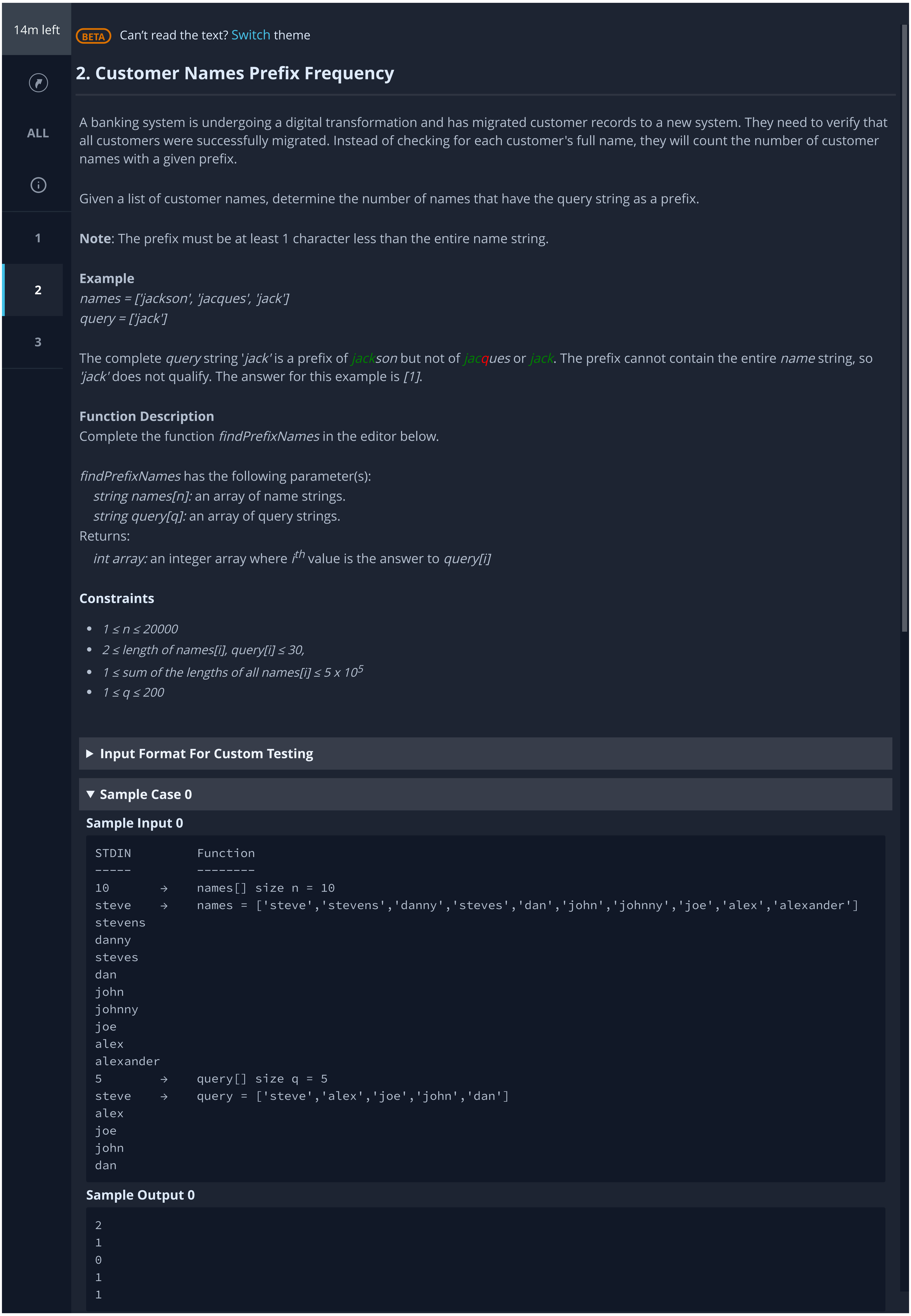

2. Customer Names Prefix Frequency

A banking system is undergoing a digital transformation and has migrated customer records to a new system. They need to verify that all customers were successfully migrated. Instead of checking for each customer's full name, they will count the number of customer names with a given prefix.

Given a list of customer names, determine the number of names that have the query string as a prefix.

Note: The prefix must be at least 1 character less than the entire name string.

Example

names = ['jackson', 'jacques', 'jack']

query = ['jack']

The complete *query* string '*jack*' is a prefix of *jackson* but not of *jacques* or *jack*. The prefix cannot contain the entire *name* string, so '*jack*' does not qualify. The answer for this example is *[1]*.

Function Description

Complete the function *findPrefixNames* in the editor below.

findPrefixNames has the following parameter(s):

string names[n]: an array of name strings.

string query[q]: an array of query strings.

Returns:

int array: an integer array where i^{th} value is the answer to *query[i]*

Constraints

- $1 \leq n \leq 20000$
- $2 \leq$ *length of names[i], query[i]* ≤ 30,
- $1 \leq$ *sum of the lengths of all names[i]* $\leq 5 \times 10^5$
- $1 \leq q \leq 200$

▶ **Input Format For Custom Testing**

▼ **Sample Case 0**

Sample Input 0

```
STDIN          Function
-----          --------
10       →     names[] size n = 10
steve    →     names = ['steve','stevens','danny','steves','dan','john','johnny','joe','alex','alexander']
stevens
danny
steves
dan
john
johnny
joe
alex
alexander
5        →     query[] size q = 5
steve    →     query = ['steve','alex','joe','john','dan']
alex
joe
john
dan
```

Sample Output 0

```
2
1
0
1
1
```