```python
import fileinput

class Node:
    def __init__(self, children=None, is_leaf=False, visited=0):
        if children is None:
            self.children = {}
        else:
            self.children = children
        self.is_leaf = is_leaf
        self.visited = visited

def add(root, name):
    node = root
    node.visited += 1
    for key in node.children:
        pre, _key, _name = extract_prefix(key, name)
        if _key == '':
            # there is a match of a part of the key
            child = node.children[key]
            return add(child, _name)
        if pre != '':
            child = node.children[key]
            # need to split
            _node = Node(
                children={_key: child},
                visited=child.visited
            )
            del node.children[key]
            node.children[pre] = _node
            return add(_node, _name)
    node.children[name] = Node(is_leaf=True, visited=1)
```

```
def find(root, name):
    node = root
    for key in node.children:
        pre, _key, _name = extract_prefix(key, name)
        if _name == '':
            return node.children[key].visited
        if _key == '':
            return find(node.children[key], _name)
    return 0

def extract_prefix(str1, str2):
    n = 0
    for c1, c2 in zip(str1, str2):
        if c1 != c2:
            break
        n += 1
    return str1[:n], str1[n:], str2[n:]

############
root = Node()

inputs = fileinput.input()
first_line = inputs.readline()
num_operations = int(first_line)

for ii in range(num_operations):
    _action, _name = inputs.readline().strip().split()
    if _action == 'add':
        add(root, _name)
    elif _action == 'find':
        print(find(root, _name))
```