[CS 11] Prac 6j – Bake Book

Problem Statement

Bakers all around the world have their own social media site Bake Book, a website where they can share their culinary creations and recipes.

The bakers responsible for this website, however, are not programmers—they're bakers—so there's little automation going on in the server. (Their *server* is actually a human being!)

Please help them out!

In the website, pairs of accounts may be *friends* or not.

They need to process *r* requests. Each request is one of three kinds:

- **Register** *x*. Given a new name *x*, make a new account with that name in the site, and then print `ok`. If the name *x* already exists, print `already registered` and do not create a new account.
- **MakeFriends** *x* *y*. Given names *x* and *y*, if *x* and *y* are not both registered names in the site, print `not found`. If *x* and *y* are registered in the site but *x = y*, print `invalid`. If both names are registered and distinct but are currently friends, print `already friends`. If they are registered, distinct, and aren't already friends, make them friends, and then print `ok`.
- **NumFriends** *x*. Given name *x*, print the number of friends of *x*. If *x* is not a registered name, print `not found`.

Task Details

Your task is to implement a function called `bakebook`. This function has a single parameter, a `tuple` of *r* requests. Each request is one of the following forms:

- `("register", x)` where `x` is a `str`.
- `("make_friends", (x, y))` where `x` and `y` are `str`s.
- `("num_friends", x)` where `x` is a `str`.

The function must not return anything; it should `print` its output.

Restrictions

(See 6a for more restrictions)

For this problem:

- Loops and lists are allowed.
- Up to 16 function definitions are allowed.
- Recursion is **disallowed**. (The recursion limit has been greatly reduced.)
- Sets and dictionaries are allowed.
- Generators and comprehensions are allowed.
- The following names are allowed: `set`, `dict`, `iter`, `next`, `any`, `all`, `popitem`, `setdefault`, `update`, `add`, `discard`.
- The source code limit is 3000.

Example Calls

Example 1 Function Call

```
bakebook((                                              Copy
    ("num_friends", "gordon"),
    ("register", "gordon"),
    ("num_friends", "gordon"),
    ("register", "ramsey"),
    ("register", "ramsay"),
    ("register", "ramses"),
    ("make_friends", ("gordon", "ramsay")),
    ("make_friends", ("ramsay", "gordon")),
    ("num_friends", "gordon"),
    ("num_friends", "ramses"),
    ("num_friends", "gordan"),
    ("register", "ramsay"),
    ("make_friends", ("gordon", "ramses")),
    ("num_friends", "gordon"),
    ("num_friends", "ramsey"),
    ("make_friends", ("gordon", "gordon")),
    ("make_friends", ("garden", "garden")),
))
```

Example 1 Output

```
not found                                               Copy
ok
0
ok
ok
ok
ok
already friends
1
0
not found
already registered
ok
2
0
invalid
not found
```

Constraints

- The function `bakebook` will be called at most 50,000 times.
- The sum of the *r*s will be ≤ 150,000.
- $0 \le r \le 150{,}000$
- Each name is a nonempty string of at most 6 lowercase English letters.

Scoring

- You get 10 ❤ points if you solve all test cases where:
  - $r \le 50$
  - the sum of *r*s is ≤ 500.
  - the output of `make_friends` and `register` are always `ok`.
  - and the output of `num_friends` is never `not found`.
- You get 10 ❤ points if you solve all test cases where:
  - $r \le 50$
  - the sum of *r*s is ≤ 500.
  - the output of `make_friends` and `register` are always `ok`.
- You get 80 ❤ points if you solve all test cases where:
  - $r \le 50$
  - the sum of *r*s is ≤ 500.
- You get 75 ❤ points if you solve all test cases where:
  - $r \le 4{,}000$
  - the sum of *r*s is ≤ 8,000.
- You get 25 💔 points if you solve all test cases.

❓ Clarifications

No clarifications have been made at this time.

✔ **Points:** 200 (partial)
⏱ **Time limit:** 6.0s
Memory limit: 1G

✎ **Author:**
kvatienza (Kevin Atienza)

❯ **Problem type**

❮ **Allowed languages**
~~NONE~~, py3