**Tìm kiếm tên theo tiền tố (phiên bản đầy đủ)**

Trong một hệ thống game online, mỗi người chơi có:

* **ID** (số nguyên dương, cố định, không thay đổi trong suốt vòng đời của người chơi).
* **Tên hiển thị** (chuỗi, có thể thay đổi).

Bạn cần xử lý các truy vấn quản lý và tìm kiếm tên người chơi.

**Các loại truy vấn**

1. **REGISTER id name**
   * Đăng ký người chơi mới với ID = id và tên = name.
   * **Điều kiện**:
     + id chưa từng tồn tại trước đó.
     + name chưa được bất kỳ người chơi nào khác sử dụng.
   * Nếu vi phạm điều kiện → in ra mã lỗi.
2. **RENAME id name**
   * Đổi tên người chơi có ID = id thành name.
   * **Điều kiện**:
     + id tồn tại.
     + name khác tên hiện tại của người chơi đó.
     + name chưa được bất kỳ người chơi nào khác sử dụng.
   * Nếu vi phạm điều kiện → in ra mã lỗi.
3. **FIND prefix**
   * In ra danh sách **tên** của tất cả người chơi **bắt đầu** bằng chuỗi prefix.
   * Danh sách được sắp **theo thứ tự từ điển ASCII**, **phân biệt chữ hoa–thường**.
   * Nếu không tìm thấy tên nào → in ra mã lỗi.

**Quy ước xử lý và xuất dữ liệu**

* Mỗi truy vấn được xử lý độc lập, theo đúng thứ tự xuất hiện.
* **Nếu truy vấn hợp lệ**:
  + REGISTER và RENAME: không in gì.
  + FIND: in danh sách tên tìm được, cách nhau bằng 1 dấu cách.
* **Nếu truy vấn vi phạm**: in ra **mã lỗi** tương ứng.
* **Tên** là chuỗi dài ≤ 20 ký tự, không chứa khoảng trắng, phân biệt hoa–thường.
* **prefix** cũng ≤ 20 ký tự, phân biệt hoa–thường.

**Tìm kiếm tên theo tiền tố (phiên bản đầy đủ)**

Trong một hệ thống game online, mỗi người chơi có:

* **ID** (số nguyên dương, cố định, không thay đổi trong suốt vòng đời của người chơi).
* **Tên hiển thị** (chuỗi, có thể thay đổi).

Bạn cần xử lý các truy vấn quản lý và tìm kiếm tên người chơi.

**Các loại truy vấn**

1. **REGISTER id name**
   * Đăng ký người chơi mới với ID = id và tên = name.
   * **Điều kiện**:
     + id chưa từng tồn tại trước đó.
     + name chưa được bất kỳ người chơi nào khác sử dụng.
   * Nếu vi phạm điều kiện → in ra mã lỗi.
2. **RENAME id name**
   * Đổi tên người chơi có ID = id thành name.
   * **Điều kiện**:
     + id tồn tại.
     + name khác tên hiện tại của người chơi đó.
     + name chưa được bất kỳ người chơi nào khác sử dụng.
   * Nếu vi phạm điều kiện → in ra mã lỗi.
3. **FIND prefix**
   * In ra danh sách **tên** của tất cả người chơi **bắt đầu** bằng chuỗi prefix.
   * Danh sách được sắp **theo thứ tự từ điển ASCII**, **phân biệt chữ hoa–thường**.
   * Nếu không tìm thấy tên nào → in ra mã lỗi.

**Quy ước xử lý và xuất dữ liệu**

* Mỗi truy vấn được xử lý độc lập, theo đúng thứ tự xuất hiện.
* **Nếu truy vấn hợp lệ**:
  + REGISTER và RENAME: không in gì.
  + FIND: in danh sách tên tìm được, cách nhau bằng 1 dấu cách.
* **Nếu truy vấn vi phạm**: in ra **mã lỗi** tương ứng.
* **Tên** là chuỗi dài ≤ 20 ký tự, không chứa khoảng trắng, phân biệt hoa–thường.
* **prefix** cũng ≤ 20 ký tự, phân biệt hoa–thường.

**Bảng mã lỗi**

|  |  |
| --- | --- |
| **Mã lỗi** | **Ý nghĩa** |
| ERROR\_DUPLICATE\_ID | ID đã tồn tại (trong REGISTER). |
| ERROR\_DUPLICATE\_NAME | Tên đã tồn tại (trong REGISTER hoặc RENAME). |
| ERROR\_ID\_NOT\_FOUND | ID không tồn tại (trong RENAME). |
| ERROR\_SAME\_NAME | Tên mới trùng với tên hiện tại của người chơi (RENAME). |
| ERROR\_NOT\_FOUND | Không có tên nào bắt đầu bằng prefix (trong FIND). |

**Input**

* Dòng 1: số nguyên Q (1 ≤ Q ≤ 2×10^5) — số lượng truy vấn.
* Q dòng tiếp theo: mỗi dòng là một truy vấn theo định dạng trên.
* id: số nguyên dương ≤ 10^9.
* name và prefix: chuỗi dài ≤ 20, không chứa khoảng trắng.

**Output**

* Theo quy tắc xử lý ở trên.

**Ví dụ**

**Input:**

9

REGISTER 101 Alice

REGISTER 102 Bob

REGISTER 103 Alicia

REGISTER 104 Bob

FIND Al

RENAME 101 Alice

RENAME 101 Ally

FIND Al

FIND Bo

**Output:**

ERROR\_DUPLICATE\_NAME

Alice Alicia

ERROR\_SAME\_NAME

Ally Alicia

Bob