Fraud Prevention

Problem Statement

At Groupon we need to take steps to detect and prevent fraudulent purchases. One form of fraud is an attempt from the same user to purchase a deal more than once using different credit card information. Given a set of orders, your task is to identify the orders that fall under this type of fraud.

An order is considered fraudulent if any of the following conditions apply:

- Two orders have the same e-mail address and deal id, but different credit card information, regardless of street address.
- Two orders have the same Address/City/State/Zip and deal id, but different credit card information, regardless of e-mail address.

Remember, people are tricky and are actively trying to get past this fraud detector. Your code must be able to handle the following tricks:

- E-mail and addresses (including city and state) are case insensitive: `bugs@bunny.com` is the same as `BuGS@BuNNy.COM` and `123 Sesame St.` is the same as `123 SeSAME st.`.
- The username portion of an e-mail address can have ignored characters. A `.` in an e-mail is flat out ignored, so `bugs1@bunny.com`, and `bugs.1@bunny.com` are the same e-mail address. A `+` in an e-mail means the plus and everything after is ignored, so `bugs@bunny.com` and `bugs+10@bunny.com` are the same e-mail address.
- Street addresses often have abbreviated words. `123 Sesame St.` and `123 Sesame Street` are the same address. `IL` and `Illinois` are the same state. For the purposes of not making this a typing problem, you can assume that the only abbreviated words you need to worry about are Street/St. and Road/Rd., and the only states you need to worry about are IL, CA (California) and NY (New York).
- Zip code may or may not contain a dash `-`.

We need this detection code to run quickly. The input file will be quite large. Hence, it is important that your code runs as fast as possible. That said, please remember that this fraud system is part of a larger system and one that might change over time, and we expect the structure of your code to reflect that fact.

Input Format

The first line contains an integer, N, denoting the number of records. This is followed by N lines containing one record each.

Each record contains the following information separated by commas:

- Order id (numeric)
- Deal id (numeric)
- Email address
- Street address
- City
- State
- Zip Code
- Credit Card #

Output Format

A single line of comma-separated fraudulent order ids in ascending order.

Sample Input

```
3
1,1,bugs@bunny.com,123 Sesame St.,New York,NY,10011,12345689010
2,1,elmer@fudd.com,123 Sesame St.,New York,NY,10011,10987654321
3,2,bugs@bunny.com,123 Sesame St.,New York,NY,10011,12345689010
```

Sample Output

```
1,2
```

Explanation

The first two orders are fraudulent, because they have the same address and deal, but different credit card information. The third order is not fraudulent because, although it shares personal information with the first order, it has the same credit card info and a different deal id.