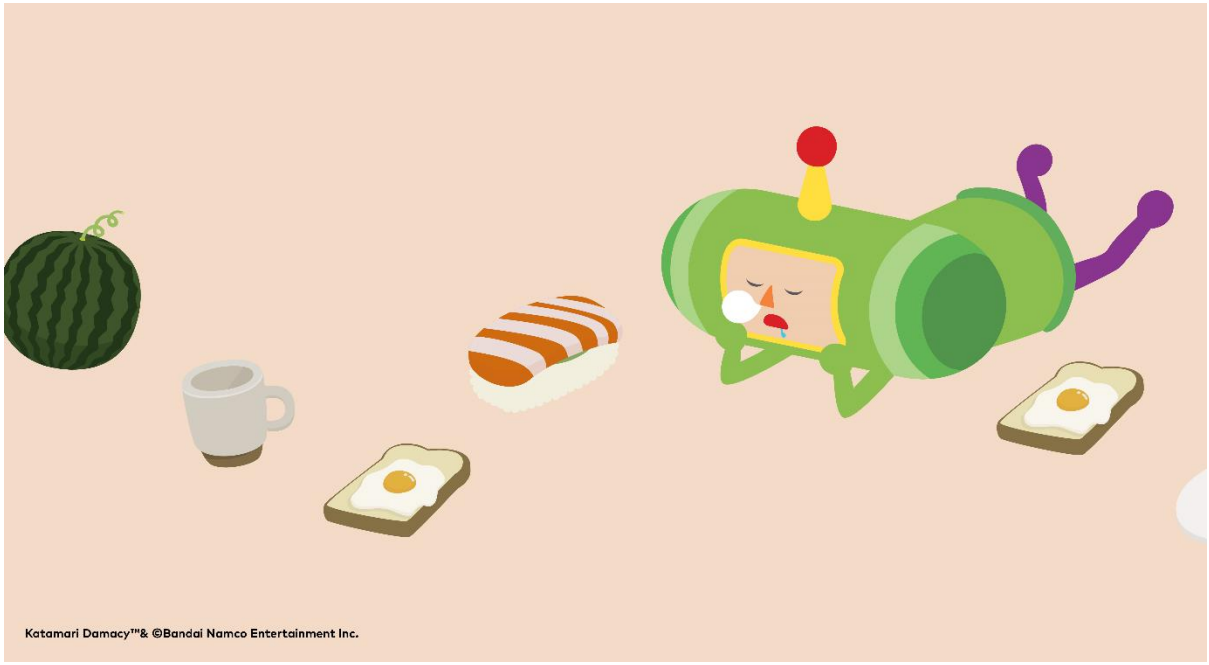
Lonely Rolling Star

By R. Rafif Aqil Aabid Hermawan

Time Limit : 1s

Memory : 2GB



Description :

Dzien' dobry! Have you ever been to Poland?

Being a prince can be a lonely thing, so the Prince is looking forward to meeting their relatives at the next meeting. To make things simpler The Prince will list a person and their friend chain to determine which one has the most relations in them so that Prince can know everyone faster!

Your job is to create a contact that stores a **person's name, ID based on order of input** and their relations with another person, which is guaranteed to be **a maximum of one.** At first there will be no relations, but over time as more people are added to the relation will grow and there will be multiple people in many relations. Let's list them all!!

ONLY USE POINTERS! ANY OTHER METHODS OF STORING RELATIONSHIP IS NOT ALLOWED!

Constraints:

- $0 \leq$ input, n ≤ 100
- 0< names, relations < 100

Input:

- First input is **n**, which is how many people to be added to the contact list.
- Next n times is the input of **name** and **relations. name** represents a new person that is being inserted to the contact list, **relations** represents the relations the current person has with other people, your program should check on the contact list on whether those person exist or not. If the person exists, then store the **pointer** to that relation person to the current one being inputted, otherwise leave it empty/NULL. A person's relation can also be NULL if the inputted relations is `"-"`

Output:

- Program should output the relations every person has. relations starts from the newest person inputted up until the first. When a person already appears in earlier relations, **do not make them a separate relations / Make them the beginning of a relation**. Further explanation will be given on examples.

Examples :

- **Example 0**

Input
1 William -
Output
---Relation 1--- William

Explanation :

William

The only person in the contact list is William, so only show that one relation.

- **Example 1**

Input
2 William - Sigourney William
Output
---Relation 1--- Sigourney -> William

Explanation :

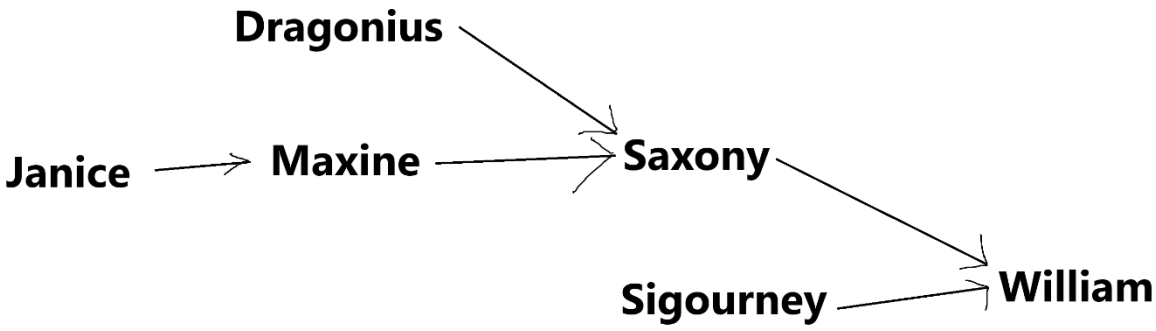
Now there are two people in the contact list and Sigourney can establish relations with William.

- **Example 2**

Input
6 William - Sigourney William Saxony William Maxine Saxony Dragonius Saxony Janice Maxine
Output
---Relation 1---

```
Janice -> Maxine -> Saxony -> William
---Relation 2---
Dragonius -> Saxony -> William
---Relation 3---
Sigourney -> William
```

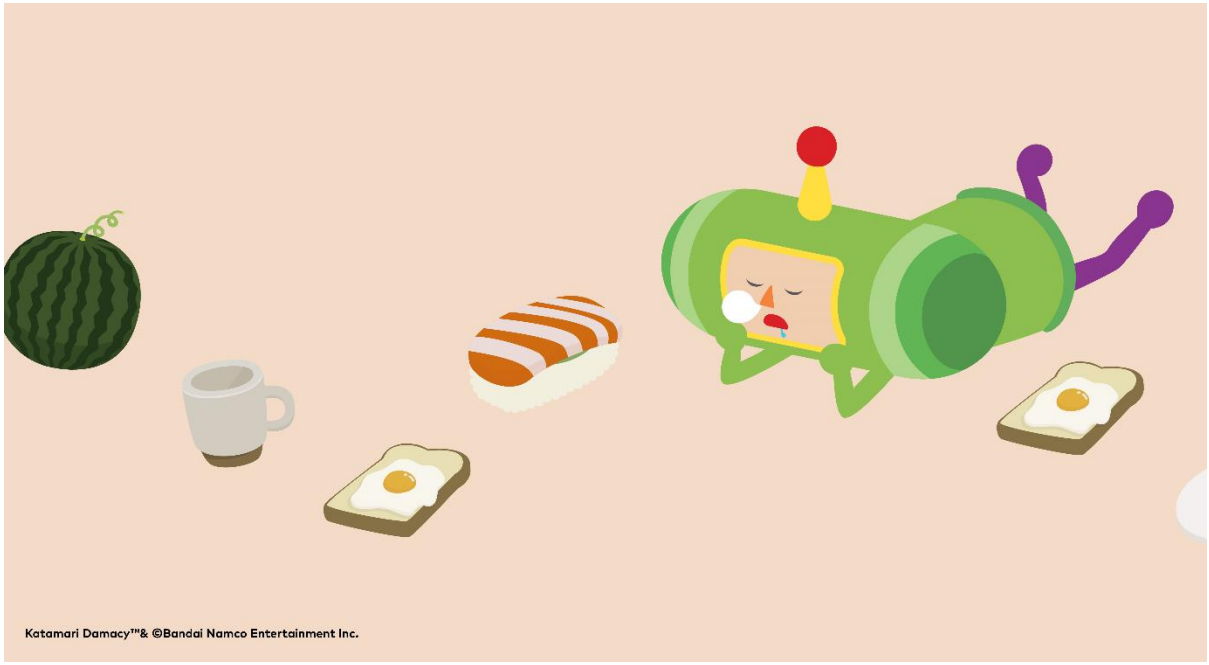
Explanation :



Multiple relationships are involved. Because Janice is the newest person in the contact their relation is being output first. Saxony has multiple relations and so does William. It is important to note that relations are determined by the first person in the relationships, therefore relations can start at Janice, Dragonius, and Sigourney with respect to the order of insertion to contact

Lonely Rolling Star

By R. Rafif Aqil Aabid Hermawan
Time Limit : 1s
Memory : 2GB



Deskripsi :

Dzien’ dobry! Pernakah kalian ke Polandia?

Menjadi seorang pangeran bisa terasa sepi, jadi Prince sangat menantikan untuk bertemu relatif dia di pertemuan selanjutnya. Untuk menyederhanakan pertemuan besok Price membuat daftar orang dan hubungan mereka agar cepat berkenalan!

Tugasmu adalah membuat kontak yang menyimpan **nama orang**, **ID berdasarkan urutan masukan**, dan hubungan dengan orang lain, yang **dijamin maksimal satu**. Pertamanya tidak akan ada relasi antar sesama, namun semakin banyak orang yang ditambahkan maka semakin besar relasi yang dibangun dan jumlah orang dalam relasi tersebut. Mari kita list semuanya!

HANYA GUNAKAN POINTER UNTUK MENYIMPAN RELASI! SELAIN METODE ITU TIDAK DIPERBOLEHKAN!

Constraints:

- 0 ≤ input, n ≤ 100
- 0< names, relations < 100

Input:

- Input pertama adalah **n**, yaitu jumlah orang yang akan dimasukkan kedalam kontak
- n selanjutnya adalah input dari **name** dan **relations. name** adalah nama orang baru yang dimasukkan kedalam kontak, **relations** adalah relasi yang dimiliki oleh orang saat ini dengan yang lain, program kalian harus mengecek didalam list kontak apakah orang tersebut ada, jika iya simpan pointer dari kontak orang tersebut ke orang yang sedang diinput, selain itu biarkan kosong/NULL. Relasi seseorang dapat dibiarkan kosong/NULL jika relasi yang diinput adalah `"-"`

Output:

- Program harus mengoutput semua relasi tiap orang punya. Relasi dimulai dari orang paling baru yang dimasukkan kedalam kontak hingga pertama. Ketika seseraong sudah pernah ada di relasi sebelumnya, jangan buat dia relasi terpisah / Membuat dia menjadi awal suatu relasi. Penjelasan lebih lanjut akan diberikan di examples.

Examples :

- **Example 0**

Input
1 William -
Output
---Relation 1--- William

Penjelasan :

William

William adalah satu-satunya orang di dalam list kontak, sehingga hanya tunjukkan satu relasi tersebut.

- **Example 1**

Input
2 William - Sigourney William
Output
---Relation 1--- Sigourney -> William

Explanation :

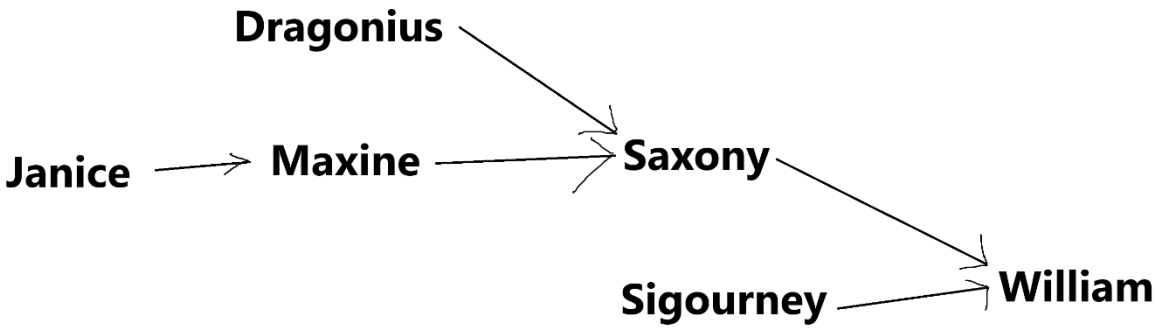
Sekarang ada dua orang didalam list kontak, dan Sigourney bisa memiliki hubungan dengan William.

- **Example 2**

Input
6 William - Sigourney William Saxony William Maxine Saxony Dragonius Saxony Janice Maxine
Output
---Relation 1---

```
Janice -> Maxine -> Saxony -> William
---Relation 2---
Dragonius -> Saxony -> William
---Relation 3---
Sigourney -> William
```

Explanation :



Ada beberapa relasi yang terjadi. Karena Janice adalah orang paling baru dalam kontak maka relation dia lah yang dioutput duluan. Bisa terlihat Saxony memiliki beberapa relasi, sama dengan William. Penting untuk dicatat kalau relasi ditentukan oleh orang pertama dalam relasi, sehingga relasi bisa dimulai dari Janice, Dragonius, dan Sigourney dengan memperhatikan urutan masuk kontak.