

F. 地底探險

Problem ID: underground

U1 時空的小 B 是一位地底世界的探險家，他常常嗑了一堆ドーパミン (dopamine，多巴胺) 後進入地底世界探險。然而小 B 有時候會因為不小心嗑了太多ドーパミン導致太嗨，讓他在地底世界中迷路，沒辦法好好在地底世界探險。於是小 B 決定告訴你他在地底世界探險的移動路徑以及他做了什麼事情，請在小 B 需要時告訴他他想要的資訊。

地底世界是由大量的地底空間及隧道構成的。每個隧道會連接兩個深度恰好差一的地底空間，其中深度的定義是從地表到達這個地底空間最少所需經過的隧道數量。每個地底空間可以被複數個隧道連接，但其中一定有唯一的一個隧道連到深度恰好為當前地底空間深度減一的空間。另外為了區別每個空間，每個空間都會被給予一個名字。小 B 在探險時可以做一些事情，例如挖一條新的隧道通往新的地底空間，使一條隧道連接著的地底空間崩塌，或是透過一個隧道走到某個相鄰的空間。簡單來說：

1. 小 B 可以挖一條隧道從目前所在的空間通往一個新的地底空間，並且小 B 會順便為此新地底空間命名。為了能夠唯一的分辨每個地底空間，同個空間連接到的深度為當前空間深度加一的地底空間都有不同的名字。
2. 小 B 可以讓一條從目前所在的空間通往深度較深的地底空間的隧道崩塌。而在此隧道崩塌後，必須通過這個隧道才能抵達的所有深度較深的地底空間都會一起崩塌。
3. 小 B 可以從目前所在的空間經過恰好一條隧道後，走到某個相鄰的空間。

如同前文所說，因為小 B 太嗨了，所以有時候小 B 會忘記他在哪個空間，或是忘記從現在所在的空間經過恰好一條隧道後能通往哪些深度較深的地底空間，又或是做出一些其他的危險事項。一旦發生了這種情況，請立刻告訴小 B 他該知道的事情。

小 B 會依序告訴你他要做的動作或是想知道的事情，每件事情的格式如下所述：

1. "dig meow"：代表小 B 從當前空間挖了一條新的隧道，且將此隧道通往的新地底空間命名為 meow。如果已經存在能經過恰好一個隧道前往且深度較深的地底空間擁有相同的名字，請輸出 "'meow' exist!"，並無視此條資料，否則不需要輸出。請自行將 meow 替換成該資料記載的名字。
2. "collapse meow"：代表小 B 讓連接著目前空間與深度較深且命名為 meow 的地底空間的隧道崩塌。如果不存在這樣的地底空間，請輸出 "'meow' not exist!"，並無視此條資料，否則不需要輸出。請自行將 meow 替換成該資料記載的名字。

3. "go to meow"：代表小 B 前往了某個恰透過一條隧道連接著，且深度比當前空間深的，且名稱為 meow 的地底空間。如果不存在這樣的地底空間，請輸出"'meow' not exist!"，並無視此條資料，否則不需要輸出。請自行將 meow 替換成該資料記載的名字。
4. "go back"：代表小 B 前往了那個唯一的，恰透過一條隧道連接著，且深度比當前空間淺的空間。如果小 B 現在已經在地表了的話，請輸出"You are on the ground!"，並無視此筆資料，否則不需要輸出。
5. "where am I"：代表小 B 忘記了自己目前在哪裡。請列出從地表開始，到達這個空間之前的所有空間的名字，再輸出當前空間的名字。如果要被印出的名字數量太多，則只需要印出最後 100 個即可。詳細格式請參考範例測試資料及 Note 部份。
6. "where can I go"：代表小 B 忘記了有哪些地底空間透過恰一條隧道連接著當前空間，且深度比當前空間深。請依照字典序輸出所有符合條件的地底空間的名字，並用恰一個空格分隔那些名字。如果符合條件的地底空間的名字超過 100 個，則只需輸出字典序前 100 小的地底空間的名字，並加上三個小數點。詳細請參考範例測試資料及 Note 部份。請不要輸出多餘的空格。

以上格式在輸入時都不包含雙引號。

Input

輸入的第一行是一個正整數 N ，代表小 B 給你了 N 筆紀錄。接下來的 N 行，每一行恰為一筆紀錄，格式如上所述。

- $1 \leq N \leq 50000$
- $1 \leq$ 每個地底空間的名字的長度 ≤ 10
- 每個地底空間的名字只包含小寫英文字母

Output

請閱讀題目敘述，輸出該輸出的東西。

由於輸出量可能會非常大，所以請將要被輸出的內容**全部**送到下面提供的函式 `AddAnswer`，並在最後呼叫 `PrintAnswer` 用來輸出答案。

除了 `PrintAnswer` 輸出的內容之外，不建議自行輸出任何東西，以免造成預期外的結果。

```
1 int HashValue = 01234567;
2 void AddAnswer(char *s) {
3     int n = strlen(s);
4     for (int i = 0; i < n; ++i)
5         HashValue = (HashValue * 13111 + s[i]) % 1000000123;
6 }
7 void PrintAnswer() {
8     printf("%d\n", HashValue);
9 }
```

Note

1. "where am I" 的輸出說明：完整輸出的範例請參考下方：

```
ground -> firstname -> secondname -> thirdname -> meow -> pog
```

請注意，第一項一定會是 `ground`。

只需要輸出後四個的輸出範例：

```
-> secondname -> thirdname -> meow -> pog
```

請留意若有名字被省略時，前面有 `->` 要輸出。

2. "where can I go" 的輸出說明：完整輸出的範例請參考下方：

```
a aa aaa aaaa bb bcd meow zoo zzz
```

只需要輸出前五個的輸出範例：

```
a aa aaa aaaa bb ...
```

請注意在最後一個名字後方有接著...。... 只有在有名字被省略時才需要輸出，例如總共只有五個名字，則不需要加上...。

另外，如果沒有任何符合條件的地底空間，也要輸出一個換行字元，詳細請參考下方的範例。

3. 以上兩點當中，實際上要輸出的數量請參考題目敘述。

範例測試資料輸出的字串原本如下：

```
ground
'd' exist!
ground -> a -> a -> a
a aa b c d
'e' not exist!
'a' not exist!

'a' not exist!
You are on the ground!
```

Sample Input 1	Sample Output 1
<pre>27 where am I where can I go dig a dig b dig c dig d dig d go to a dig a go to a dig a go to a where am I go back go back go back dig aa where can I go collapse e collapse a go to a dig a go to a where can I go go to a go back go back</pre>	<pre>603303985</pre>

This page is intentionally left blank.