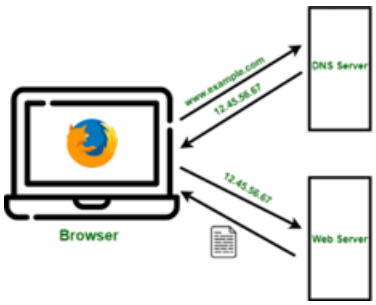
Problem . DNS

Time limit: 1 seconds

Khi Nga cố gắng truy cập một website có tên miền như " **example.com**", máy tính sẽ gửi một yêu cầu tìm kiếm đến máy chủ DNS (Domain Name System) để xác định địa chỉ IP tương ứng với tên miền đó. Quá trình này gọi là "phân giải tên miền".



Để cải thiện thời gian tìm và truy xuất địa chỉ IP của website, cache DNS ra đời, nó sẽ lưu trữ các thông tin IP của các website vào bộ nhớ cache của máy tính người dùng. Trong quá trình phân giải tên miền, đầu tiên máy tính sẽ tìm kiếm trong bộ nhớ cache DNS của chính nó để xem có tồn tại địa chỉ IP tương ứng với tên miền đã được lưu trữ trước đó hay không. Nếu không có bản ghi nào được lưu trữ trong bộ nhớ cache DNS thì máy tính mới gửi một yêu cầu tìm kiếm đến máy chủ DNS. Máy chủ DNS sẽ tìm kiếm trong cơ sở dữ liệu và trả về IP tương ứng. Sau đó, 1 bản ghi về tên miền và IP tương ứng sẽ được lưu trữ vào cache DNS để tăng tốc độ truy cập những lần sau. Tất nhiên tên miền không tồn tại sẽ không được lưu vào cache DNS. Khi bộ nhớ cache DNS đầy, thì phải loại bỏ bớt bản ghi để giải phóng bộ nhớ cho các bản ghi mới. Có nhiều thuật toán có thể được sử dụng, trong số đó LRU (Least Recently Used) là một thuật toán phổ biến. Các bản ghi được lưu trữ theo thứ tự truy cập, khi bộ nhớ cache DNS đầy thì các bản ghi ít được truy cập hơn sẽ được loại bỏ; giả sử máy tính của Nga sử dụng thuật toán này để triển khai cho hệ thống cache DNS.

Tuy nhiên có một kỹ thuật tấn công mạng nhằm thay đổi thông tin trong bộ nhớ cache của một máy chủ DNS hoặc một máy khách nhằm chuyển hướng các yêu cầu truy cập đến địa chỉ IP sai, đó là DNS Poisoning (hay còn gọi là DNS Spoofing hoặc DNS Cache Poisoning). Kỹ thuật tấn công này thường được thực hiện bằng cách thay đổi các bản ghi DNS trong bộ nhớ cache của một máy chủ DNS hoặc một máy khách, hoặc bằng cách giả mạo các gói tin DNS và gửi chúng đến các máy chủ DNS. Khi người dùng cố gắng truy cập một tên miền nhất định, máy chủ DNS sẽ trả về một địa chỉ IP sai và chuyển hướng người dùng đến trang web giả mạo hoặc các địa chỉ IP độc hại. Nhận thấy DNS Poisoning là một trong những kỹ thuật tấn công phổ biến nhất hiện nay, và có thể gây ra những hậu quả nghiêm trọng cho người dùng và các tổ chức, bạn Nga quyết định tìm hiểu về kỹ thuật tấn công này và muốn xây dựng một chương trình đơn giản kiểm tra xem người dùng có khả năng bị tấn công bởi DNS Poisoning hay không.

Chương trình sẽ xử lý một tập dữ liệu gồm danh sách các truy vấn thuộc một trong hai loại sau:

- **1 d1 ip2** : Cập nhật domain **d1** có địa chỉ IP là **ip2** tại máy chủ DNS, giả sử ban đầu cơ sở dữ liệu tại máy chủ rỗng

- **2 d3** : Máy khách gửi yêu cầu tìm kiếm địa chỉ IP của domain **d3** đến máy chủ DNS

Vì chương trình còn đơn giản nên Nga chỉ có thể kiểm tra được cache DNS ở tại máy khách, và thay vì tìm kiếm trên cache DNS trước, chương trình gửi yêu cầu đến máy chủ DNS trước, nếu IP trả về khác với IP được lưu trong cache DNS thì có khả năng đã bị tấn công DNS Poisoning.

Hãy dự đoán chương trình của Nga sẽ in ra gì các bạn nhé.

Input

- Dòng thứ nhất chứa 2 số nguyên n, m lần lượt là số lượt truy vấn và số bản ghi tối đa cache có thể lưu

trữ. Với $0 < n, m \leq 10^5$.

- n dòng sau, mỗi dòng sẽ thuộc một trong hai loại truy vấn mô tả ở trên, trong đó **d1**, **ip2** và **d3** là dữ liệu đúng và kích thước không quá 20 kí tự.

Output

In ra dòng cảnh báo "**Warning: Possible DNS Poisoning detected!**" nếu tồn tại IP do máy chủ DNS gửi về khác với IP được lưu trong cache DNS, ngược lại in ra "**Everything looks ok!**".

Examples

standard input	standard output
7 2 1 example.com 192.168.1.2 2 example.com 1 abc.net 192.168.4.5 2 abc.net 1 abc.net 192.168.1.4 1 helloworld.edu 1.2.3.4 2 abc.net	Warning: Possible DNS Poisoning detected!
9 2 1 example.com 192.168.1.2 2 example.com 1 abc.net 192.168.4.5 2 abc.net 1 helloworld.edu 1.2.3.4 1 abc.net 192.168.1.4 2 helloworld.edu 2 example.com 2 abc.net	Everything looks ok!

Explanations

Ở ví dụ 2, khi người dùng truy vấn tên miền **example.com** nó sẽ được nạp vào cache, tiếp theo sẽ là **abc.net**.

Cache DNS	
example.com	192.168.1.2

Cache DNS	
example.com	192.168.1.2
abc.net	192.168.4.5

Lúc này cache đã đủ 2 bản ghi nên khi người dùng truy vấn **helloworld.edu**, tên miền được truy vấn xa nhất sẽ được xóa đi là **example.com**; sau đó, **helloworld.edu** được nạp vào cache.

Cache DNS	
abc.net	192.168.4.5
helloworld.edu	1.2.3.4

Tương tự như vậy khi người dùng truy cập **example.com** thì **abc.net** bị xóa đi và thay thế bằng **example.com**.

Cache DNS	
helloworld.edu	1.2.3.4
example.com	192.168.1.2

Đến lần truy cập **abc.net** cuối cùng trong cache đã không còn bản ghi của abc.net để so sánh nữa nên chương trình không thể phát hiện ra khả năng bị tấn công.

Cache DNS	
example.com	192.168.1.2
abc.net	192.168.1.4