Running MoM

The bad thing about being an International Man of Mystery (IMoM) is there’s usually someone who wants to kill you. Sometimes you have to stay on the run just to stay alive. You have to think ahead. You have to make sure you don’t end up trapped somewhere with no escape.

Of course, not all IMoMs are blessed with a great deal of intelligence. You are going to write a program to help them out. You are going to make sure our IMoM knows what cities are safe to visit and which are not. It’s not enough to just be able to run (or fly) for one or two days, we have to guarantee that the IMoM can keep running for as long as might be necessary. Given a list of regular, daily flights between pairs of cities, you are going to make sure our IMoM never gets stuck in a city from which there is no escape. We’ll say there is an escape from some location if there is an infinitely long sequence of cities the IMoM could fly to making one flight each day. The IMoM never even wants to go to a city from which there isn’t an escape.

Input

Input will start with a number, \( n \), giving the number of daily flights there are between pairs of cities. This will be followed by \( n \) lines, each line containing a pair of city names. A line containing the name \( C_o \) followed by \( C_d \) indicates that there is a flight from city \( C_o \) to city \( C_d \) every day. Flights are one-way. A line like “Dallas Baltimore” means that there is a flight from Dallas to Baltimore every day. You can’t expect to fly from Baltimore back to Dallas unless there is another line that says “Baltimore Dallas”. City names are sequences of non-whitespace characters and are separated by whitespace. There are no flights that are destined for the same city from which they originate.

The description of daily flights will be followed by a list of city names.

Output

Your job is to examine the list of city names at the end and determine whether or not there is an escape from the each one. For each, you will output the name of the city, followed by the word “safe” if there is an escape and “trapped” if there is no escape.

Sample Input

5
Arlington San_Antonio
San_Antonio Baltimore
Baltimore New_York
New_York Dallas
Baltimore Arlington
San_Antonio
Baltimore
New_York

Sample Output

San_Antonio safe
Baltimore safe
New_York trapped