

12. Lauren

Program Name: Lauren.java

Input File: lauren.dat

Lauren loves to travel. She just got her brand new passport, and she is excited to travel the world. She is overwhelmed by the number of cities she wants to visit. To be the most efficient, she wants to be sure to visit every city on her list, but she wants to make sure to do it covering the least amount of distance as possible. Oh, and she doesn't really care about getting back home after reaching the last city...at least for a while.

Input: The first integer will represent the number of data sets to follow. The next line will have an integer representing the number of cities she wants to visit on this trip, followed by a list of cities, each on one line and will contain no spaces. The first city in this list represents the city that Lauren is currently in. The next line will be an integer representing the number of available flights. The following lines will be two cities, separated by a space, and an integer representing the number of miles that flight covers.

Output: For each data set, print the names of the cities in the order that she should visit them. She must start in the city that she is currently in. Each flight is represented by a "=>" sign between the cities.

Assumptions: There will always be a flight to the city Lauren wants to visit. If there is a flight from city A to city B there is assumed to be the same flight from city B to city A. If there is no flight listed between two cities it is assumed she cannot fly between them. There will be at least two cities to travel between. There will always exist a route between any two cities, although it may take several hops. The distance between any two adjacent cities will be greater than 1 mile. The best solution will never require Lauren to backtrack.

Sample Input:

```
3
3
SanFrancisco
NewYork
LosAngeles
3
SanFrancisco LosAngeles 383
SanFrancisco NewYork 2905
LosAngeles NewYork 2789
5
Austin
Istanbul
Calcutta
NewYork
London
10
Austin Istanbul 6500
Austin Calcutta 8773
Austin NewYork 1513
Austin London 4921
Istanbul Calcutta 3652
Istanbul NewYork 5020
Istanbul London 1556
Calcutta NewYork 7929
Calcutta London 4853
NewYork London 3465
4
Reykjavik
Aarhus
Geneva
Budapest
5
Reykjavik Aarhus 1515
Reykjavik Geneva 2379
Reykjavik Budapest 2443
Aarhus Budapest 933
Geneva Budapest 787
```

Sample Output:

```
SanFrancisco => LosAngeles => NewYork
Austin => NewYork => London => Istanbul => Calcutta
Reykjavik => Aarhus => Budapest => Geneva
```