

**Introduction**

In a recent prophetic dream, your team earned a trip to the UIL finals. However, the penny-pinching principal at your school won't authorize the trip to Austin until you can prove that your team will be responsible and choose the most economical route to UT Austin.

In response, your team is writing software that will scan Texas highway maps and determine the shortest route. The other members of the team are writing the OCR software, and you are authoring the module that will process city connection and road length data to determine the shortest path between any two specified towns, which should also be the path that uses the least amount of valuable school-funded gasoline.

**Input**

Input to this problem will consist of a (non-empty) series of up to 100 data sets. Each data set will be formatted according to the following description, and there will be **no blank lines** separating data sets:

1. *Dataset Start Line* – A single line, "DATASET  $n$ ", where  $1 \leq n \leq 10$  and  $n$  is the number of lines in this data set (not including the *Dataset Start Line*).
2. *Town List* – A single line containing a number ( $m \mid 2 \leq m \leq 10$ ) followed by a space-separated list of  $m$  town names. The first name will be your starting town, and the second name will be your destination. All other towns are possible stops along your route. Each name will be from one to twenty alphabetical characters in length.
3. *Road List* – A series of  $(n - 1)$  lines representing the lengths of known roads between pairs of towns from the *Town List*. Each line will list two town names followed by the length of a known road ( $d \mid 1 \leq d \leq 1000$ ) connecting them. A given pair of towns will appear in this list at most once and possibly never.

**Output**

Output for each data set will consist of two parts. The first is an exact replica of the *Dataset Start Line*. The second will be the list of towns visited followed by the total distance traversed. The towns should be listed in the order they would be visited when traveling from the starting town to the destination by the shortest possible path. You may assume that there will be one and only one shortest path.

**Example: Input File**

```
DATASET 5
4 Dallas Austin Houston SanAntonio
Dallas SanAntonio 275
Dallas Houston 238
SanAntonio Austin 79
Houston Austin 162
DATASET 2
2 MyHouse YourHouse
MyHouse YourHouse 1
DATASET 4
3 MyHouse YourHouse Somewhere
MyHouse YourHouse 300
YourHouse Somewhere 100
MyHouse Somewhere 100
```

**Output to screen**

```
DATASET 5
Dallas SanAntonio Austin 354
DATASET 2
MyHouse YourHouse 1
DATASET 4
MyHouse Somewhere YourHouse 200
```