

Algorithms Fundamentals with Python: Exam

Please submit your solutions (source code) to all the below-described problems in [Judge](#).

2. Critical Neighborhoods

Get ready for a road trip adventure! You are given a list of **roads that connect different cities**. Each road has a **distance** associated with it. But wait, there's a catch - **some roads are closed**! Your task is to plan a route from a given **start city** to a given **end city** that takes into account the closed roads and **covers the minimum possible distance**.

You need to write a program that takes the input as described below and produces the output as described below. Can you find the best route and complete your journey, despite the obstacles in your way?

Input

- The first line contains an integer **r** - the number of roads.
- The next **r** lines each contain a bidirectional road in the format:
`"{city1} - {city2} - {distance}"`.
- The next line contains a comma-separated list of closed roads in the format:
`"{city1}-{city2},{city3}-{city4},..."`
where each closed road is represented by a tuple of two distinct city names.
- The next line contains a string **start_city** representing the name of the city to start the journey.
- The last line contains a string **end_city** representing the name of the city to end the journey.

Output

- Print the path from the start to the end city in the following format:
`"{city1} - {city2} - {city3} ... {cityN}"`
- Print the total distance from the start to the end city.

Constraints

- All city names are non-empty strings containing only uppercase letters (A-Z).
- The distance between two cities along a road is a positive integer in the range **[1... 100]**.
- The number of roads will be in the range **[1... 10]**.
- The number of cities will be in the range **[1... 50]**.
- The start and end cities are distinct and are both present in the list of roads.
- There will always be a road from the start to the end city.

Examples

| Input | Output |
|--|--------------------|
| 8 A - B - 2 A - C - 4 B - C - 1 B - D - 4 C - D - 3 C - E - 5 D - E - 2 B - E - 1 B-D A E | A - B - E 3 |
| 8 A - B - 2 A - C - 4 B - C - 1 B - D - 4 C - D - 3 C - E - 5 D - E - 2 B - E - 1 B-E A E | A - B - C - E 8 |