# Algorithms Advanced with C#: Exam

Please submit your solutions (source code) to all the below-described problems in [Judge](https://judge.softuni.org/Contests/4059).

## 2. Social Media Tracker

You are given a social network represented as a **collection of users**, each with a unique identifier. Each user can have **one or more friends**, and each friend relationship has a **non-negative integer value** that represents the **strength of the friendship** between the users. The higher the integer value, the stronger the friendship between the users.

Your task is to develop an algorithm that will take two user IDs as input and output a path between them with the highest total strength. In other words, the algorithm should find the path with the best sum of friendship strengths between the two users.

However, there is a catch: if there are multiple paths with the **same total strength value**, you should select the one with **the** **least number of hops**. When we say **"friend hops"**, we are referring to the **number of friendships** that need to be traversed to reach from one user to the other.

### Input

* + The first line of the input contains an integer **r** (1 ≤ **r** ≤ 1000) - the number of friend relationships.
  + The next **r** lines each contain three space-separated values: **"{userA} {userB} {influence}"**, where **userA** and **userB** are unique user identifiers and influence is a non-negative integer (0 ≤ influence ≤ 10^6).
    - These values represent a friendship between **userA** and **userB**, with the strength of the friendship being represented by the influence value.
    - The influence is in the **userA** -> **userB** direction.
  + On the following line, you will receive the **startUser**.
  + On the last line, you will receive the **destinationUser**.

### Output

* + Print the best sum of friendship strengths and the hops in the following format: **"({bestSum}, {hops})"**.

### Constraints

* + There will be always a path from the **startUser** to the **destinationUser**.
  + There will be no case with multiple paths with equal **bestSum** and **hops**.

### Examples

|  |  |
| --- | --- |
| **Input** | **Output** |
| 5  P G 5  G I 10  P A 3  A N 15  N G 3  P  I | (31, 4) |
| 5  P G 21  G I 10  P A 3  A N 15  N G 3  P  I | (31, 2) |