# C++ Fundamentals – Regular Exam – 10 February 2024

Please submit your source code to all below-described problem in [Judge](https://judge.softuni.org/Contests/4435/CPlusPlus-Fundamentals-Regular-Exam-10-December-2023).

## Space Travel

You're one of the science officers of a space carrier: a huge ship, which visits unknown solar systems and obtains astronomical data about them. You must write a program, which will analyze the live data of solar's system orbiting bodies and will be able to execute commands and return data results.

### Input

Your program receives the following input data:

1. The names of the bodies of the system, each one on a single row. If the name equals "END", that means all the names of the solar system are received.
   1. The first name is **always** the name of the Star of the solar system.
   2. Names of the bodies will **never** contain a space.
   3. There will be no more than 20 bodies in each solar system, including the main star.
2. After the names, you will receive a square matrix of integers, which gives the "time to travel" between each one of the bodies (in hours).
   1. Due to specifics of the space travel, travelling from A to B may take 40 hours, but at the same time travelling from B to A may take 24 hours, e.g. the matrix is not symmetrical!
3. After the travel data, you will receive a number of rows with names of the bodies, each separated with space. Each row describes a path in the system, which passes around each one of the bodies.
4. If you read a row that says “END”.

### Output

1. After you read the travel time matrix (point 2 above), find out and print:
   1. The shortest time to travel between two single points in the system, e.g. from which point to which point would take the least time to travel.
   2. The longest time to travel between two single points in the system.
   3. **Note:** there might be more then one shortest and/or longest routes. In this case you must print all of them.
2. After you read each row of travel itinerary (point 3 above), calculate the total time, needed to travel for the row, and print it out on a single row
3. After the input finishes (after the last END), print out the sum of all travel routes.

**Hints:**

1. You will need to remember the order, in which your bodies come.
2. You will need a square matrix of integers, in which the index of each element is the index of the body in the initial list. Remember how to find an element’s index in a vector (the input vector of all bodies), and use that index in the matrix.

### Example 1

|  |  |
| --- | --- |
| **Input** | **Explanation** |
| Betelgeuse  Planet1  Planet2  Planet3  END  0 2 5 7  2 0 4 9  3 1 0 5  9 4 3 0  Planet3 Planet2  Betelgeuse Planet1 Planet3  Planet3 Planet2 Planet3 Planet1 Betelgeuse  END | The system contains the star Betelgeuse and three planets Planet1, Planet2, Planet3.  The time to travel is from Betelgeuse to each of the planets: 2 5 7 (the first row of the matrix), from Planet1 to each other body 2 4 9 (the second row of the matrix, etc.  Please note the zeroes in the diagonal: this is because the time to travel from each body to itself is 0 (we’re already there).  After the time matrix we have three routes:   1. “Planet3 Planet2” 2. “Betelgeuse Planet1 Planet3” 3. “Planet3 Planet2 Planet3 Planet1 Betelgeuse” |
| **Output** | **Explanation** |
| 1: Planet2 -> Planet1  9: Planet1 -> Planet3  9: Planet3 -> Betelgeuse | The shortest route is from Planet2 to Planet1 (1).  The longest routes (9) are from Planet1 Planet3, and from Planet3 to Betelgeuse. |
| 3  11  14  28 | The three routes have the following lengths:   1. “Planet3 Planet2” is **3**: 2. “Betelgeuse Planet1 Planet3” is 2 + 9 = **11**: 3. “Planet3 Planet2 Planet3 Planet1 Betelgeuse” takes 3 + 5 + 4 + 2 = **14**   After you receive the last END, the sum of all the routes is 3 + 11 + 14 = **28**. |

### Example 2

|  |  |
| --- | --- |
| **Input** | **Output** |
| Tatoo  Tatooine  Planet3  PlanetX  END  0 3 7 4  2 0 4 2  5 1 0 9  3 8 11 0  Tatooine Planet3 PlanetX  PlanetX Tatooine  END | 1: Planet3 -> Tatooine  11: PlanetX -> Planet3  13  8  21 |

### Example 3

|  |  |
| --- | --- |
| **Input** | **Output** |
| Astra  Nebulo  Quasar  Vortex  Pulsar  Galaxia  Cosmos  END  0 4 8 5 7 3 7  1 0 9 1 3 7 4  5 2 0 7 8 3 1  7 9 8 0 2 5 8  3 3 1 5 0 9 7  3 5 3 1 5 0 7  2 9 5 6 4 7 0  Astra Nebulo Quasar  Vortex Pulsar Galaxia  Cosmos Astra Nebulo Quasar Vortex  Pulsar Cosmos Galaxia Astra  Nebulo Quasar Cosmos Vortex Pulsar  END | 1: Nebulo -> Astra  1: Nebulo -> Vortex  1: Quasar -> Cosmos  1: Pulsar -> Quasar  1: Galaxia -> Vortex  9: Nebulo -> Quasar  9: Vortex -> Nebulo  9: Pulsar -> Galaxia  9: Cosmos -> Nebulo  13  11  22  17  18  81 |