Base and Edge test cases

1. Description: Shortest path between two nodes connected directly (Adjacent Nodes)
   Input: Source: Location A, Target: Location B
   Expected Output: Time: 13 minutes, Path: Location A -> Location B
   Actual Output:

```
Graph successfully loaded!

Location A -> Location B(13)  Location D(4)
Location B -> Location C(7)  Location G(16)  Location D(5)
Location C -> Location D(15)
Location D -> Location E(4)  Location F(6)
Location E -> Location F(5)  Location I(8)
Location F -> Location G(20)
Location G -> Location H(6)  Location C(20)
Location H -> Location I(14)
Location I -> Location J(4)  Location G(4)  Location C(12)
Location J -> Location K(19)  Location B(10)  Location A(15)
Location K -> Location L(9)
Location L -> Location A(4)

Enter source location: Location A
Enter target location: Location B
Shortest time from Location A to Location B: 13 minutes
Path: Location A -> Location B
```

2. Description: Shortest path including multiple edges
   Input: Source: Location A, Target: Location I
   Expected Output: Time: 16 minutes, Path: Location A -> Location D -> Location E -> Location I
   Actual Output:

```
Graph successfully loaded!

Location A -> Location B(13)  Location D(4)
Location B -> Location C(7)  Location G(16)  Location D(5)
Location C -> Location D(15)
Location D -> Location E(4)  Location F(6)
Location E -> Location F(5)  Location I(8)
Location F -> Location G(20)
Location G -> Location H(6)  Location C(20)
Location H -> Location I(14)
Location I -> Location J(4)  Location G(4)  Location C(12)
Location J -> Location K(19)  Location B(10)  Location A(15)
Location K -> Location L(9)
Location L -> Location A(4)

Enter source location: Location A
Enter target location: Location I
Shortest time from Location A to Location I: 16 minutes
Path: Location A -> Location D -> Location E -> Location I
```

3. Description: Same Source and Target
   Input: Source: Location A, Target : Location A
   Expected Output: Time: 0 minutes, Path: Location A
   Actual Output:

```
Graph successfully loaded!

Location A -> Location B(13)  Location D(4)
Location B -> Location C(7)  Location G(16)  Location D(5)
Location C -> Location D(15)
Location D -> Location E(4)  Location F(6)
Location E -> Location F(5)  Location I(8)
Location F -> Location G(20)
Location G -> Location H(6)  Location C(20)
Location H -> Location I(14)
Location I -> Location J(4)  Location G(4)  Location C(12)
Location J -> Location K(19)  Location B(10)  Location A(15)
Location K -> Location L(9)
Location L -> Location A(4)

Enter source location: Location A
Enter target location: Location A
Shortest time from Location A to Location A: 0 minutes
Path: Location A
```

4. Description: Shortest path between two nodes connected directly (Non-adjacent nodes)
   Input: Source: Location A, Target : Location D
   Expected Output: Time: 4 minutes, Path: Location A -> Location D
   Actual Output:

```
Graph successfully loaded!

Location A -> Location B(13)  Location D(4)
Location B -> Location C(7)  Location G(16)  Location D(5)
Location C -> Location D(15)
Location D -> Location E(4)  Location F(6)
Location E -> Location F(5)  Location I(8)
Location F -> Location G(20)
Location G -> Location H(6)  Location C(20)
Location H -> Location I(14)
Location I -> Location J(4)  Location G(4)  Location C(12)
Location J -> Location K(19)  Location B(10)  Location A(15)
Location K -> Location L(9)
Location L -> Location A(4)

Enter source location: Location A
Enter target location: Location D
Shortest time from Location A to Location D: 4 minutes
Path: Location A -> Location D
```

5. Description: Invalid source or target location
   Input: Source: Location Z, Target : Location A
   Expected Output: Invalid source or target location
   Actual Output:

```
Graph successfully loaded!

Location A -> Location B(13)  Location D(4)
Location B -> Location C(7)  Location G(16)  Location D(5)
Location C -> Location D(15)
Location D -> Location E(4)  Location F(6)
Location E -> Location F(5)  Location I(8)
Location F -> Location G(20)
Location G -> Location H(6)  Location C(20)
Location H -> Location I(14)
Location I -> Location J(4)  Location G(4)  Location C(12)
Location J -> Location K(19)  Location B(10)  Location A(15)
Location K -> Location L(9)
Location L -> Location A(4)

Enter source location: Location Z
Enter target location: Location A
Invalid source or target location.
```

6. Description: JSON file does not exist
   Input: Changed the location of JSON file
   Expected Output: JSON file not found
   Actual Output:

```
JSON file not found!
Could not load graph.
```

7. Description: Having a source or target which does not have any incoming or outgoing edge
   Input: Source: Location A, Target : Location M (Added new node M to data)
   Expected Output : No path found
   Actual Output:

```
Graph successfully loaded!

Location A -> Location B(13)  Location D(4)
Location B -> Location C(7)  Location G(16)  Location D(5)
Location C -> Location D(15)
Location D -> Location E(4)  Location F(6)
Location E -> Location F(5)  Location I(8)
Location F -> Location G(20)
Location G -> Location H(6)  Location C(20)
Location H -> Location I(14)
Location I -> Location J(4)  Location G(4)  Location C(12)
Location J -> Location K(19)  Location B(10)  Location A(15)
Location K -> Location L(9)
Location L -> Location A(4)
Location M ->

Enter source location: Location A
Enter target location: Location M
No path found.
```

8. Description: Two different paths with same total time available between two nodes
   Input: Added new node M and edges from C to M with 5 weight and from H to M with weight 19 in the data
   Expected Output: Location F -> Location G -> Location H -> Location M
   Actual Output:

```
Graph successfully loaded!

Location A -> Location B(13)  Location D(4)
Location B -> Location C(7)  Location G(16)  Location D(5)
Location C -> Location D(15)  Location M(5)
Location D -> Location E(4)  Location F(6)
Location E -> Location F(5)  Location I(8)
Location F -> Location G(20)
Location G -> Location H(6)  Location C(20)
Location H -> Location I(14)  Location M(19)
Location I -> Location J(4)  Location G(4)  Location C(12)
Location J -> Location K(19)  Location B(10)  Location A(15)
Location K -> Location L(9)
Location L -> Location A(4)
Location M ->

Enter source location: Location F
Enter target location: Location M
Shortest time from Location F to Location M: 45 minutes
Path: Location F -> Location G -> Location H -> Location M
```