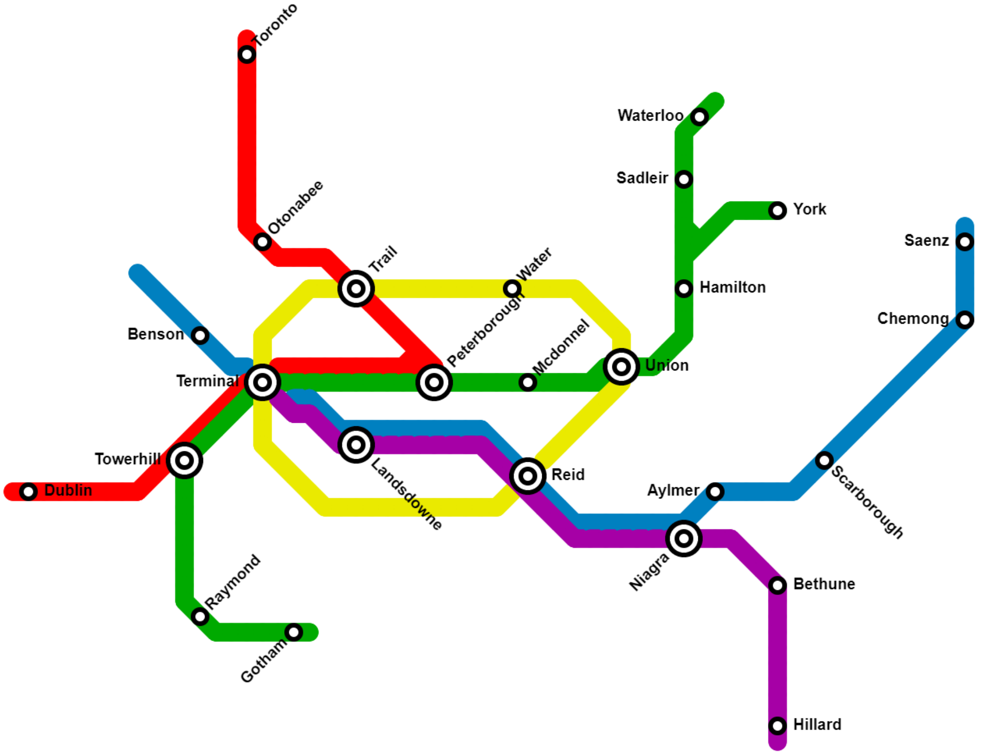
COIS 3020, Winter 2023

Subway Shortest Path

Contents

SubwayMap



Created with MetroMapMaker.com
Remix this map! Go to https://metromapmaker.com/?map=3d2O2yh8

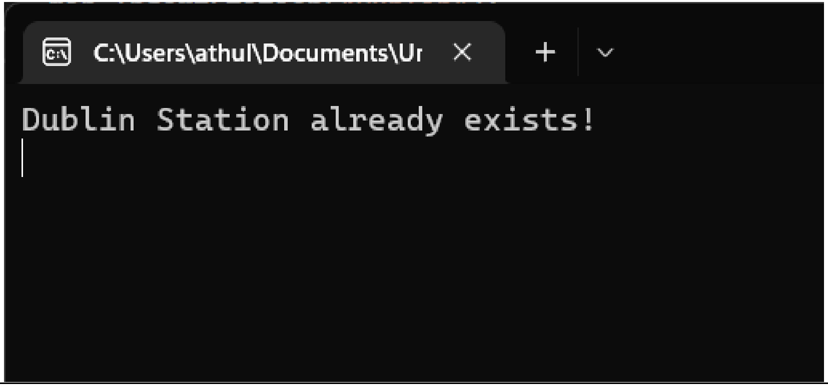
Testing

Case 1: Inserting same station twice.

Input:

```
map.InsertStation("Dublin");
map.InsertStation("Dublin");
```

Output:



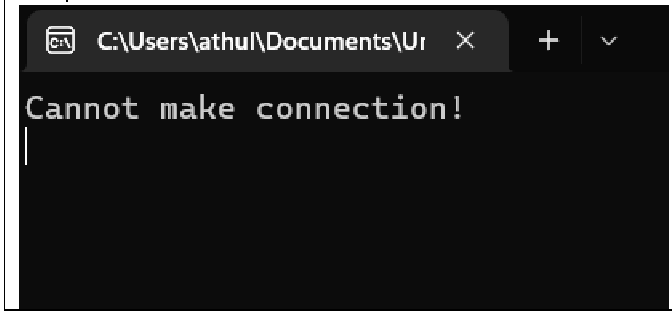
```
Dublin Station already exists!
```

Case 2: Connection between non-existed stations

Input:

```
bool val;
        val = map.InsertConnection("Oshawa", "Toronto", Colour.Red);
if (!val)
           Console.WriteLine("Cannot make connection!");
```

Output:



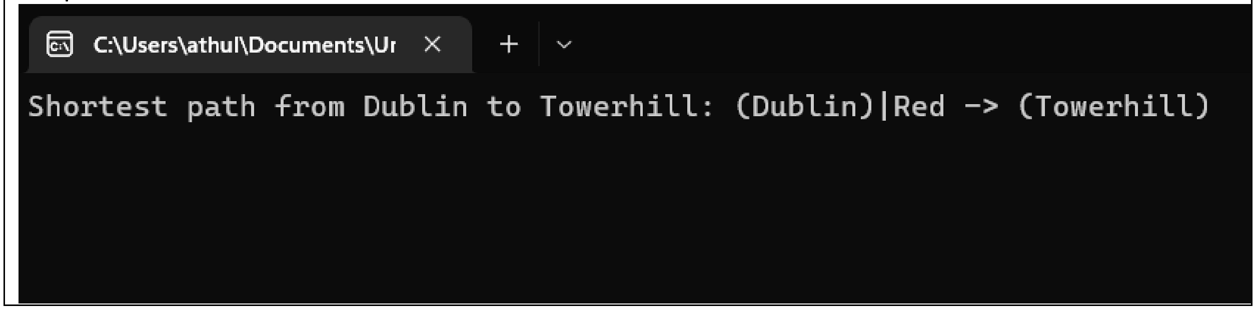
```
Cannot make connection!
```

Case 3: Shortest route between two adjacent stations

Input:

```
map.ShortestRoute("Dublin", "Towerhill");
```

Output:



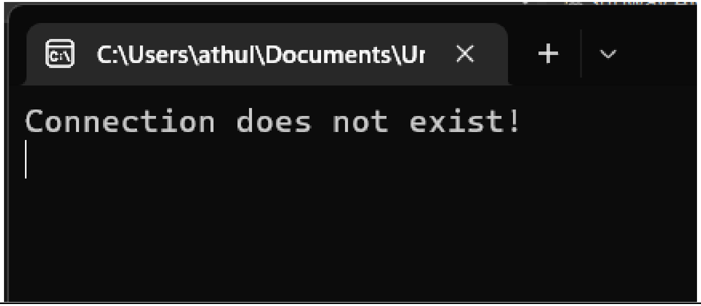
```
Shortest path from Dublin to Towerhill: (Dublin)|Red -> (Towerhill)
```

Case 4: Shortest route to the same station

Input:

```
map.ShortestRoute("Dublin", "Dublin");
```

Output:



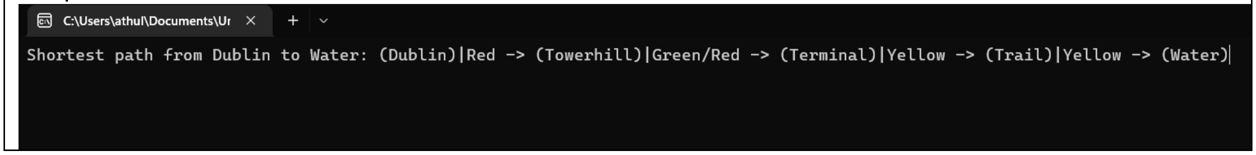
```
Connection does not exist!
```

Case 5: Shortest route between two stations with multiple routes.

Input:

```
map.ShortestRoute("Dublin", "Water");
```

Output:



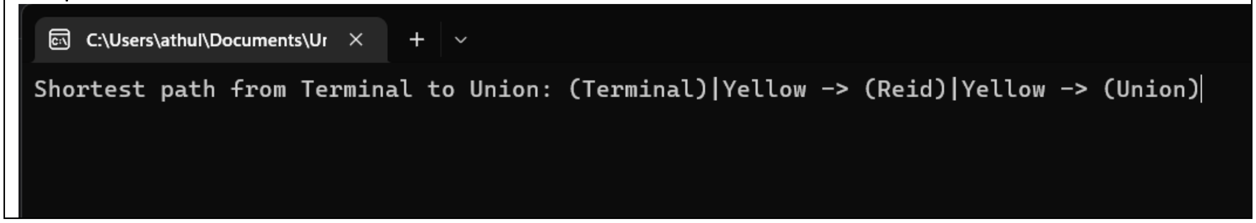
```
Shortest path from Dublin to Water: (Dublin)|Red -> (Towerhill)|Green/Red -> (Terminal)|Yellow -> (Trail)|Yellow -> (Water)
```

Case 6: Shortest route between two stations with multiple routes of same distance.

Input:

```
map.ShortestRoute("Terminal", "Union");
```

Output:



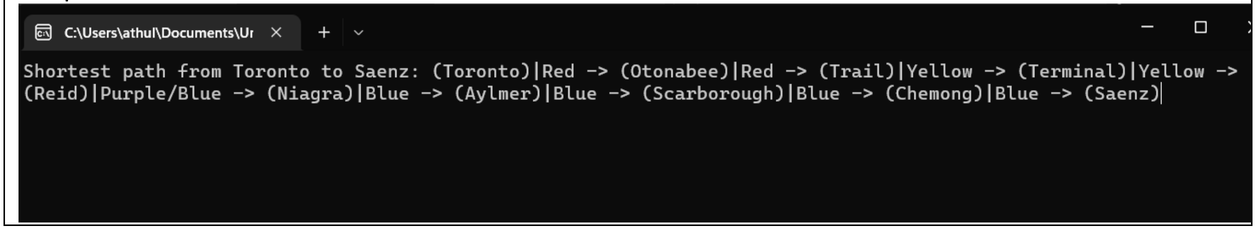
```
Shortest path from Terminal to Union: (Terminal)|Yellow -> (Reid)|Yellow -> (Union)
```

Case 7: Shortest route between two far stations.

Input:

```
map.ShortestRoute("Toronto", "Saenz");
```

Output:



```
Shortest path from Toronto to Saenz: (Toronto)|Red -> (Otonabee)|Red -> (Trail)|Yellow -> (Terminal)|Yellow ->
(Reid)|Purple/Blue -> (Niagra)|Blue -> (Aylmer)|Blue -> (Scarborough)|Blue -> (Chemong)|Blue -> (Saenz)
```

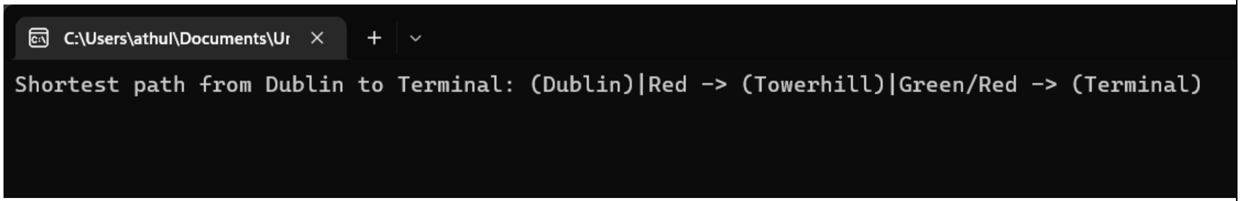
Case 8: Shortest route after removing a connection.

Input:

```
map.RemoveConnection("Dublin", "Towerhill", Colour.Red);
map.ShortestRoute("Dublin", "Terminal");
```

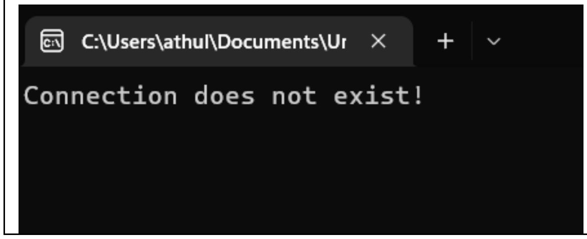
Output:

Before:



```
Shortest path from Dublin to Terminal: (Dublin)|Red -> (Towerhill)|Green/Red -> (Terminal)
```

After:



```
Connection does not exist!
```

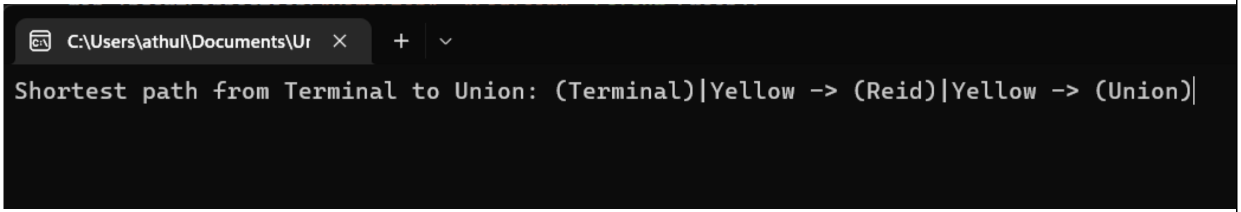
Case 9: Shortest route after removing one connection from stations connected by multiple lines.

Input:

```
map.RemoveConnection("Terminal", "Reid", Colour.Yellow);
map.ShortestRoute("Terminal", "Union");
```

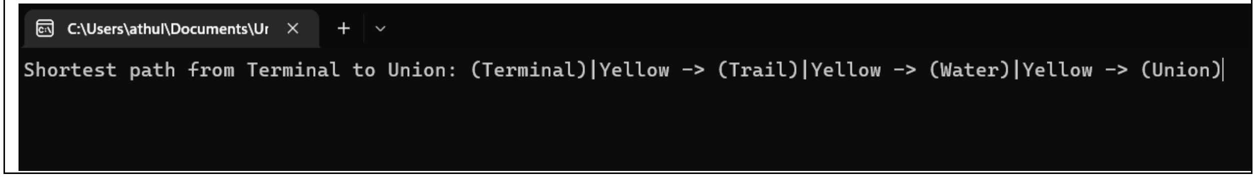
Output:

Before:



```
Shortest path from Terminal to Union: (Terminal)|Yellow -> (Reid)|Yellow -> (Union)
```

After:



```
Shortest path from Terminal to Union: (Terminal)|Yellow -> (Trail)|Yellow -> (Water)|Yellow -> (Union)
```

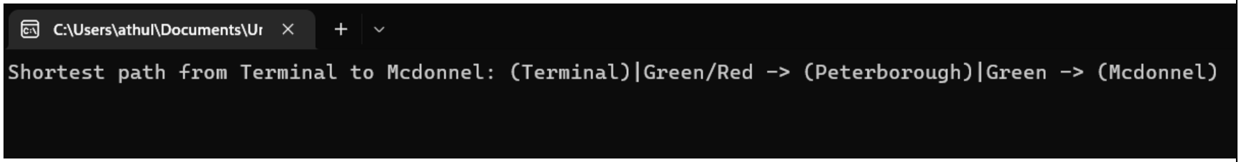
Case 10: Shortest route between two stations after removing a station in the route.

Input:

```
map.RemoveStation("Peterborough");
map.ShortestRoute("Terminal", "Mcdonnel");
```

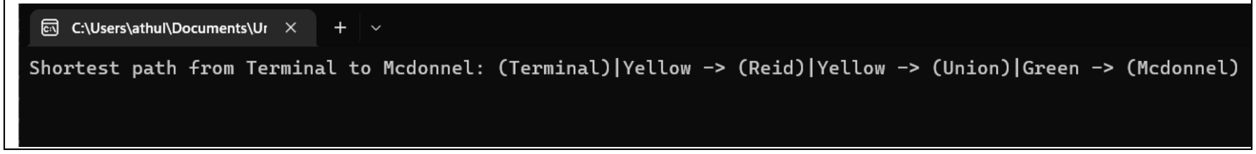
Output:

Before:



```
Shortest path from Terminal to Mcdonnel: (Terminal)|Green/Red -> (Peterborough)|Green -> (Mcdonnel)
```

After:



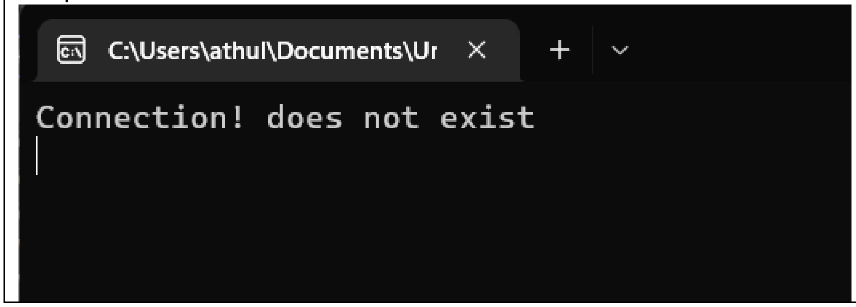
```
Shortest path from Terminal to Mcdonnel: (Terminal)|Yellow -> (Reid)|Yellow -> (Union)|Green -> (Mcdonnel)
```

Case 11: Removing connection between two stations but with the wrong colour.

Input:

```
bool val;
          val = map.RemoveConnection("Trail", "Terminal", Colour.Blue);
          if (!val)
             Console.WriteLine("Connection! does not exist");
```

Output:



```
Connection! does not exist
```