Design Decisions:

The concepts of graph data structure is used for implementation.

A 5X5 adjacency matrix created and updated according to the existence of path between 2 towns.

A large integer value (named as INFINITY) in the matrix shows, there is no path between 2 towns.

The distance between 2 routes can be calculated easily with the help of the adjacency matrix.

A recursive DFS algorithm ( with modifications to add bidirectional paths) is used to generate all the paths between 2 towns.

To generate all the paths between a cyclic trip(C-C-)

* one step back tracking to find the previous nodes before the final destination (e.g B and D)
* Use the above recursive algorithm to get all paths between C-B and C-D and then append the start (node C) at the end of all the generated paths.

The shortest path and the number of stops are calculated after generating the possible path ( Cyclic and Acyclic)

The number of routes with minimum distance is calculated by

* Generate all possible paths
* Calculate the distance of each path and compare with the given minimum distance.
* In the case of Cyclic path
  + Concatenate the possible paths by appending each string(path) in all the 3 possible ways( start , end, start and end) to generate all possible routes.
  + Multiply the route distance with different values to check whether it is within the given distance. ( from 4 –given Minimum distance because the first 3 are already generated by the above step).

The solution gives proper output for the given input in text file. The txt file is placed inside src Folder.

Also few Junit tests generated to check the solution with sample inputs.

Solution was developed using eclipse IDE ( the src folder attached).