Programming assignments

**Part 1: Undirected graphs**

*For this part you should assume that the edges defined by the vertex pairs are two-way.*

1. Write a program based on DFS which can answer questions of the type: "Find the a path from X to Y" Which should result in a list of vertices traversed from X to Y if there is a path.
2. Change the program to use BFS.
3. Write a program which can find the shortest path from vertex A to C passing trough an intermediate vertex C if such a path exists. If the path exists the program should print the path.

**Part 2: Directed graphs**

*For this part you should assume that the edges defined by the vertex pairs in the data base are one-way.*

1. Write a program that can answer if there is a path between any to vertices.
2. Write a program (i.e. devise an algorithm) which can find if a starting vertex is part of a cycle (loop) or not. If there is a cycle/loop then the program should output the vertices of the loop. (You only nee to find one cycle in case there are more than one loop) Use graphs without self-loops as input.
3. (2p) Implement a program which allows the user to find the shortest path between two nodes in a graph possibly passing through a third node. I.e. the user should be able to ask questions like:

* Which is the shortest path from A to B passing through C?

The program should output an ordered list of the nodes to traverse from A to B if such a path exists. If no such path exists then the program should output that no such path exists.

Use NYC.txt as input when not executing tests (in the case that the tests should be executed you may use another input). This is the undirected road network of New York City. The graph contains 264346 vertices and 733846 edges. It is connected, contains parallel edges, but no self-loops. The edge weights are travel times and are strictly positive - for this assignment you do not use the travel times/edge weights but assume they are equal (1).