

### Assignment 3

#### Suggested test cases (not the marking test cases)

---

Dear Students,

Hello! In order to help your work on assignment 3, I present the test cases that will use to mark your assignment 3 here. Please note that I may use another dataset to test your work, but will basically follow the following 13 test cases. I have also made a VPL for you to submit and test your work.

Please feel free to let me know if you have any doubt in the test case requirements. I am very happy to help you ☺

Best regards,

Kit.

You can check your program according to the following test cases:

1. We will call the `InsertNode` function in ascending order of the node ID. The following input is the same as the running example used in the specification.



Input 1	(Run the executable) InsertNode 0 Library_Building InsertNode 1 Hui_Oi_Chow_Science_Building InsertNode 2 University_Street InsertNode 3 Kadoorie_Biological_Sciences_Building InsertNode 4 Haking_Wong_Building InsertNode 5 Chow_Yei_Ching_Building
Output 1	(No screen output, the nodes are inserted into the graph.)

2. We will call the `InsertNode` function in any random order of the node ID.

Input 2	(Run the executable) InsertNode 1 Hui_Oi_Chow_Science_Building InsertNode 5 Chow_Yei_Ching_Building InsertNode 0 Library_Building InsertNode 3 Kadoorie_Biological_Sciences_Building InsertNode 2 University_Street InsertNode 4 Haking_Wong_Building
Output 2	(No screen output, the nodes are inserted into the graph.)

3. We will call the `InsertNode` function with node ID not starting from 0.

Input 3	(Run the executable) <pre> InsertNode 10 Hui_Oi_Chow_Science_Building InsertNode 22 Chow_Yei_Ching_Building InsertNode 32 Library_Building InsertNode 25 Kadoorie_Biological_Sciences_Building InsertNode 11 University_Street InsertNode 9 Haking_Wong_Building </pre>
Output 3	(No screen output, the nodes are inserted into the graph.)

4. We will call the `InsertNode` function with duplicate node ID, the node will not be inserted into the graph. The `InsertNode` function will output "ID exists.", followed by an `endl` on screen. The program should continue to process any upcoming commands after displaying the error message.

Input 4	(Run the executable) <pre> InsertNode 0 Library_Building InsertNode 1 Hui_Oi_Chow_Science_Building InsertNode 2 University_Street InsertNode 3 Kadoorie_Biological_Sciences_Building InsertNode 4 Haking_Wong_Building InsertNode 5 Chow_Yei_Ching_Building InsertNode 5 Main_Building </pre>
Output 4	ID exists.

- Note that in the above test case the graph is built for the first 6 calls of `InsertNode` function.
- The program is still active to receive upcoming commands.

5. We will call the `InsertEdge(x, y)` function with both `x` and `y` exist in the graph.

Input 5	Assume that we continue with the inputs of test case 1. <pre> InsertEdge 0 1 InsertEdge 1 0 InsertEdge 1 2 InsertEdge 2 1 InsertEdge 0 3 InsertEdge 3 0 InsertEdge 2 4 InsertEdge 4 2 InsertEdge 3 4 InsertEdge 4 3 InsertEdge 4 5 InsertEdge 5 4 </pre>
Output 5	(No screen output, the edges are inserted into the graph.)

6. We will call the `InsertEdge(x, y)` function with either `x` or `y`, or both does not exists in the graph. `InsertEdge(x, y)` should output "No such node." once, followed by an `endl` on screen.

Input 6a	Assume that we continue with the inputs of test case 1. <code>InsertEdge 100 1</code>
Output 6a	No such node.
Input 6b	Assume that we continue with the inputs of test case 1. <code>InsertEdge 1 100</code>
Output 6b	No such node.
Input 6c	Assume that we continue with the inputs of test case 1. <code>InsertEdge 100 100</code>
Output 6c	No such node.

7. We will call the `CommonNeighbor(x, y)` function where node `x` and `y` have common neighbors. If there are more than one common neighbors, output them in ascending order of the node ID, line by line.

Input 7	Assume that we continue with the inputs of test case 5. <code>InsertEdge 1 3</code> <code>InsertEdge 3 1</code> <code>CommonNeighbor 2 3</code>
Output 7	1 Hui_Oi_Chow_Science_Building 4 Haking_Wong_Building

8. We will call the `CommonNeighbor(x, y)` function where node `x` and `y` do not have common neighbors. The function outputs "No common neighbor.", followed by an `endl` on screen.

Input 8	Assume that we continue with the inputs of test case 5. <code>CommonNeighbor 1 5</code>
Output 8	No common neighbor.

9. We will call the `CommonNeighbor(x, y)` function where node `x` and `y` are the same.

Input 9	Assume that we continue with the inputs of test case 5. <code>CommonNeighbor 0 0</code>
Output 9	1 Hui_Oi_Chow_Science_Building 3 Kadoorie_Biological_Sciences_Building

10. We will call the `CommonNeighbor(x, y)` function with `x` or `y` or both does not exists in the graph. `CommonNeighbor(x, y)` should output "No such node." once, followed by an `endl` on screen.

Input 10a	Assume that we continue with the inputs of test case 1. <code>CommonNeighbor 100 0</code>
Output 10a	No such node.
Input 10b	Assume that we continue with the inputs of test case 1. <code>CommonNeighbor 0 100</code>
Output 10b	No such node.
Input 10c	Assume that we continue with the inputs of test case 1. <code>CommonNeighbor 100 100</code>
Output 10c	No such node.

11. We will call the `ShortestPath(x, y)` function where node `x` and `y` have a path to return. If there are more than one shortest paths, output any one of them.

Input 11a	Assume that we continue with the inputs of test case 5. <code>ShortestPath 0 4</code>
Output 11a	0 Library_Building 3 Kadoorie_Biological_Sciences_Building 4 Haking_Wong_Building
Input 11b	Assume that we continue with the inputs of test case 5. <code>ShortestPath 1 5</code>
Output 11b	1 Hui_Oi_Chow_Science_Building 2 University_Street 4 Haking_Wong_Building 5 Chow_Yei_Ching_Building
Input 11c	Assume that we continue with the inputs of test case 5. <code>ShortestPath 5 0</code>
Output 11c	5 Chow_Yei_Ching_Building 4 Haking_Wong_Building 3 Kadoorie_Biological_Sciences_Building 0 Library_Building

12. We will call the `ShortestPath(x,y)` function where node `x` and `y` are not connected in the graph (i.e., there are no path to reach from `x` to `y` in the graph.). In this case, `ShortestPath(x,y)` outputs "No path found.", followed by an endl on screen.

Input 12a	Assume that we continue with the inputs of test case 5. InsertNode 6 University_Hall ShortestPath 0 6
Output 12a	No path found.
Input 12b	Assume that we continue with the inputs of test case 5. InsertNode 7 Clinical_Pathology_Building InsertNode 8 Faculty_of_Medicine_Building InsertEdge 7 8 InsertEdge 8 7 ShortestPath 8 0
Output 12b	No path found.
Input 12c	Assume that we continue with the inputs of test case 5. InsertNode 7 Clinical_Pathology_Building InsertNode 8 Faculty_of_Medicine_Building InsertEdge 7 8 InsertEdge 8 7 InsertEdge 3 8 ShortestPath 7 0
Output 12c	No path found.
Input 12d (say, if we have a one way shuttle from KBS to Medicine building)	Assume that we continue with the inputs of test case 5. InsertNode 7 Clinical_Pathology_Building InsertNode 8 Faculty_of_Medicine_Building InsertEdge 7 8 InsertEdge 8 7 InsertEdge 3 8 ShortestPath 0 7
Output 12d	0 Library_Building 3 Kadoorie_Biological_Sciences_Building 8 Faculty_of_Medicine_Building 7 Clinical_Pathology_Building

13. We will call the `ShortestPath(x,y)` function where node `x` and `y` are the same node.

Input 13a	Assume that we continue with the inputs of test case 5. ShortestPath 0 0
Output 13a	0 Library_Building
Input 13b	Assume that we continue with the inputs of test case 5. ShortestPath 4 4
Output 13b	4 Haking_Wong_Building

I wish you like this assignment and get a good grade in this task ☺ - Kit