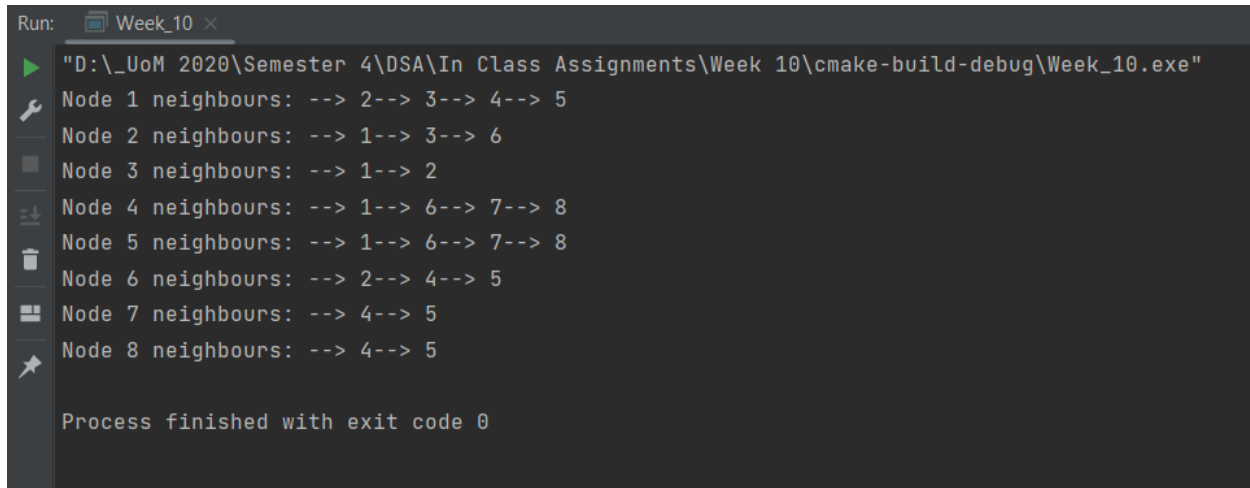


Github link - <https://github.com/StupidME2000/Sem-4-DSA-ICA-Week-10.git>

Section 1



```
Run: Week_10 x
"D:\_UoM 2020\Semester 4\DSA\In Class Assignments\Week 10\cmake-build-debug\Week_10.exe"
Node 1 neighbours: --> 2--> 3--> 4--> 5
Node 2 neighbours: --> 1--> 3--> 6
Node 3 neighbours: --> 1--> 2
Node 4 neighbours: --> 1--> 6--> 7--> 8
Node 5 neighbours: --> 1--> 6--> 7--> 8
Node 6 neighbours: --> 2--> 4--> 5
Node 7 neighbours: --> 4--> 5
Node 8 neighbours: --> 4--> 5

Process finished with exit code 0
```

To modify the `addEdge` function to accept directed edges instead of undirected edges, we can remove the step of adding both `(u, v)` and `(v, u)` to the adjacency lists. Instead, we will only add the edge `(u, v)` to the adjacency list of `u`. Here's the modified `addEdge` function for a directed graph:

```
void addEdge(int u, int v) {
    nodes[u - 1].neighbours.push_back(v);
}
```

With this adjustment, an edge with the notation `(u, v)` will only be added to the adjacency list of `u` when it is added. This change indicates that the graph has a directed edge running from `u` to `v`.

Section 2

- First let's look at Node 1 neighbors.

Node 1 : [2, 3, 4, 5]

- Other than Node 4 it has 3 more neighbors. Which are;

- Node 2 : [1, 3, 6]

- Node 3 : [1, 2]

- Node 5 : [1, 6, 7, 8]

$$Sim(a,b) = \frac{\# \text{ of shared neighbours between a,b}}{\text{Total neighbours in a,b}} = \frac{a \cap b}{a \cup b}$$

Node 4 and 2 :

$$Sim(4,2) = \frac{[1,6,7,8] \cap [1,3,6]}{[1,6,7,8] \cup [1,3,6]} = 2/5 = 0.4$$

Node 4 and 3 :

$$Sim(4,3) = \frac{[1,6,7,8] \cap [1,2]}{[1,6,7,8] \cup [1,2]} = 1/5 = 0.2$$

Node 4 and 5 :

$$Sim(4,5) = \frac{[1,6,7,8] \cap [1,6,7,8]}{[1,6,7,8] \cup [1,6,7,8]} = 4/4 = 1$$

- Let's consider Node 4's neighbors. Node 4 : [1, 6, 7, 8] We got the highest similarity value for 4 and 5. I suggest Node 5 for Node 4 to become friends