
Started on	Saturday, 8 November 2025, 3:18 PM
-------------------	------------------------------------

State	Finished
--------------	----------

Completed on	Saturday, 8 November 2025, 3:35 PM
---------------------	------------------------------------

Time taken	16 mins 39 secs
-------------------	-----------------

Grade	80.00 out of 100.00
--------------	----------------------------

Question **1**

Not answered

Mark 0.00 out of
20.00

Write a Python program with Function to insert the string values into Circular Queue.

For example:

Input	Result
3 java python c++	['java', 'python', 'c++']
2 java C++ Python	Queue is full ['java', 'C++']

Answer: (penalty regime: 0 %)

1 ||

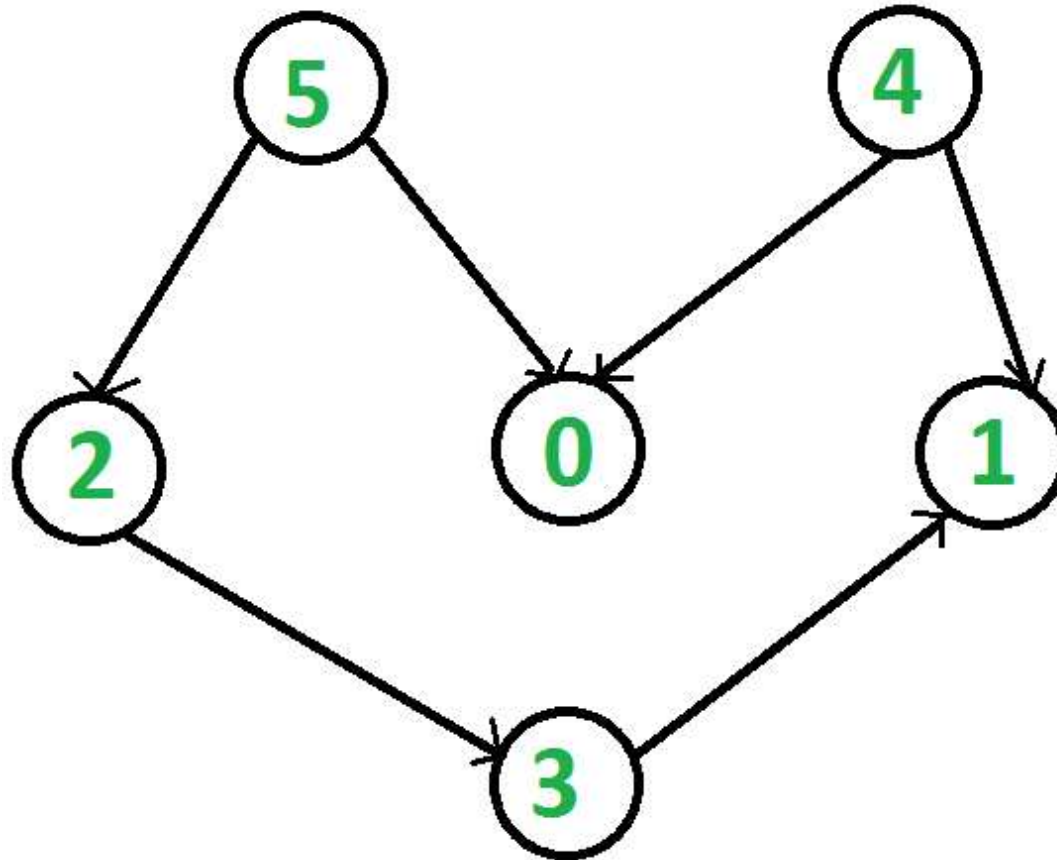
11

Question **2**

Correct

Mark 20.00 out of 20.00

Write a Python program to print topological sorting of a DAG



For example:

Result

Topological Sort of the given graph is
5 4 2 3 1 0

Answer: (penalty regime: 0 %)

Reset answer

```
1 # A Python3 program to print topological sorting of a DAG
2 def addEdge(u, v):
3     global adj
4     adj[u].append(v)
5
6 # The function to do DFS() and stores departure time
7 # of all vertex
8 def DFS(v):
9     global visited, departure, time
10    visited[v] = 1
11    for i in adj[v]:
12        if visited[i] == 0:
13            DFS(i)
14    departure[time] = v
15    time += 1
16
17 # The function to do Topological Sort. It uses DFS().
18 def topologicalSort():
19     for i in range(V):
20         if (visited[i]==0):
21             DFS(i)
22
```

	Expected	Got	
✓	Topological Sort of the given graph is 5 4 2 3 1 0	Topological Sort of the given graph is 5 4 2 3 1 0	✓

Passed all tests! ✓

Correct

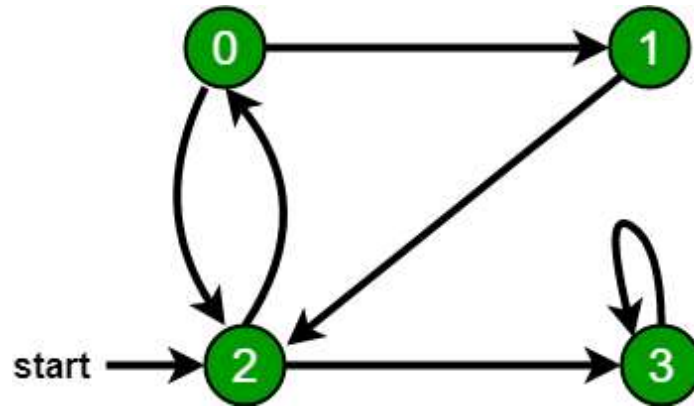
Marks for this submission: 20.00/20.00.

Question 3

Correct

Mark 20.00 out of 20.00

Write a Python program to print [BFS](#) traversal from a given source vertex.



Note : write [BFS](#) function only.

For example:

Input	Result
2	Following is Breadth First Traversal (starting from vertex 2) 2 0 3 1

Answer: (penalty regime: 0 %)

Reset answer

```
1 # Python3 Program to print BFS traversal
2 # from a given source vertex. BFS(int s)
3 # traverses vertices reachable from s.
4 from collections import defaultdict
5
6 # This class represents a directed graph
7 # using adjacency list representation
8 class Graph:
9
10     # Constructor
11     def __init__(self):
12
13         # default dictionary to store graph
```

```

13         # default dictionary to store graph
14         self.graph = defaultdict(list)
15
16         # function to add an edge to graph
17     def addEdge(self,u,v):
18         self.graph[u].append(v)
19
20     # Function to print a BFS of graph
21     def BFS(self, s):
22

```

	Input	Expected	Got	
✓	2	Following is Breadth First Traversal (starting from vertex 2) 2 0 3 1	Following is Breadth First Traversal (starting from vertex 2) 2 0 3 1	✓
✓	1	Following is Breadth First Traversal (starting from vertex 1) 1 2 0 3	Following is Breadth First Traversal (starting from vertex 1) 1 2 0 3	✓

Passed all tests! ✓

Correct

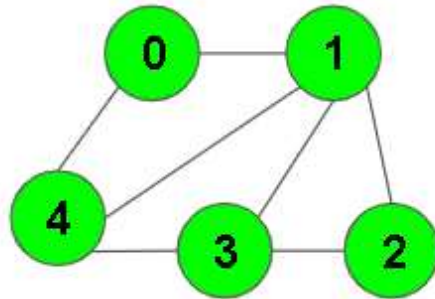
Marks for this submission: 20.00/20.00.

Question **4**

Correct

Mark 20.00 out
of 20.00

Write A Python program to demonstrate the adjacency
list representation of the graph.



Note : Write Print function only.

For example:

Result

Adjacency list of vertex 0

0 -> 4 -> 1

Adjacency list of vertex 1

1 -> 4 -> 3 -> 2 -> 0

Adjacency list of vertex 2

2 -> 3 -> 1

Adjacency list of vertex 3

3 -> 4 -> 2 -> 1

Adjacency list of vertex 4

4 -> 3 -> 1 -> 0

Answer: (penalty regime: 0 %)

Reset answer


```
2 | A Python program to demonstrate the adjacency
3 | list representation of the graph
4 | """
5 |
6 | # A class to represent the adjacency list of the node
7 |
8 |
9 | class AdjNode:
10 |     def __init__(self, data):
11 |         self.vertex = data
12 |         self.next = None
13 |
14 |
15 | # A class to represent a graph. A graph
16 | # is the list of the adjacency lists.
17 | # Size of the array will be the no. of the
18 | # vertices "V"
19 | class Graph:
20 |     def __init__(self, vertices):
21 |         self.V = vertices
22 |         self.graph = [None] * self.V
```

	Expected	Got	
✓	Adjacency list of vertex 0 0 -> 4 -> 1	Adjacency list of vertex 0 0 -> 4 -> 1	✓
	Adjacency list of vertex 1 1 -> 4 -> 3 -> 2 -> 0	Adjacency list of vertex 1 1 -> 4 -> 3 -> 2 -> 0	
	Adjacency list of vertex 2 2 -> 3 -> 1	Adjacency list of vertex 2 2 -> 3 -> 1	
	Adjacency list of vertex 3 3 -> 4 -> 2 -> 1	Adjacency list of vertex 3 3 -> 4 -> 2 -> 1	
	Adjacency list of vertex 4 4 -> 3 -> 1 -> 0	Adjacency list of vertex 4 4 -> 3 -> 1 -> 0	

Passed all tests! ✓

Correct

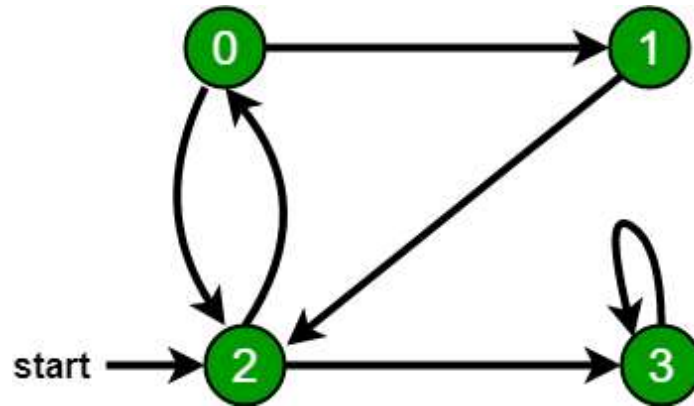
Marks for this submission: 20.00/20.00.

Question 5

Correct

Mark 20.00 out of 20.00

Write a Python program to print [DFS](#) traversal from a given source vertex.



Note : write DFSUTIL function only.

For example:

Input	Result
2	Following is DFS from (starting from vertex 2) 2 0 1 3

Answer: (penalty regime: 0 %)

Reset answer

```
1 # Python3 program to print DFS traversal
2 # from a given graph
3 from collections import defaultdict
4
5 # This class represents a directed graph using
6 # adjacency list representation
7
8
9 class Graph:
10
11     # Constructor
12     def __init__(self):
13
```

```

14         # default dictionary to store graph
15         self.graph = defaultdict(list)
16
17         # function to add an edge to graph
18     def addEdge(self, u, v):
19         self.graph[u].append(v)
20
21         # A function used by DFS
22     def DFSUtil(self, v, visited):

```

	Input	Expected	Got	
✓	2	Following is DFS from (starting from vertex 2) 2 0 1 3	Following is DFS from (starting from vertex 2) 2 0 1 3	✓
✓	1	Following is DFS from (starting from vertex 1) 1 2 0 3	Following is DFS from (starting from vertex 1) 1 2 0 3	✓

Passed all tests! ✓

Correct

Marks for this submission: 20.00/20.00.