A SOWJANYA

RA1911030010081

AI EXPERIMENT – 4

 Implementation and Analysis of DFS and BFS

AIM:

To implement DFS and BFS

CODE DFS:

graph **=** {

'A' : ['B','C'],

'B' : ['D'],

'C' : ['F'],

'D' : ['E', 'F'],

'E' : [],

'F' : ['A']

}

visited **=** set()

**def** dfs(visited, graph, node):

**if** node **not** **in** visited:

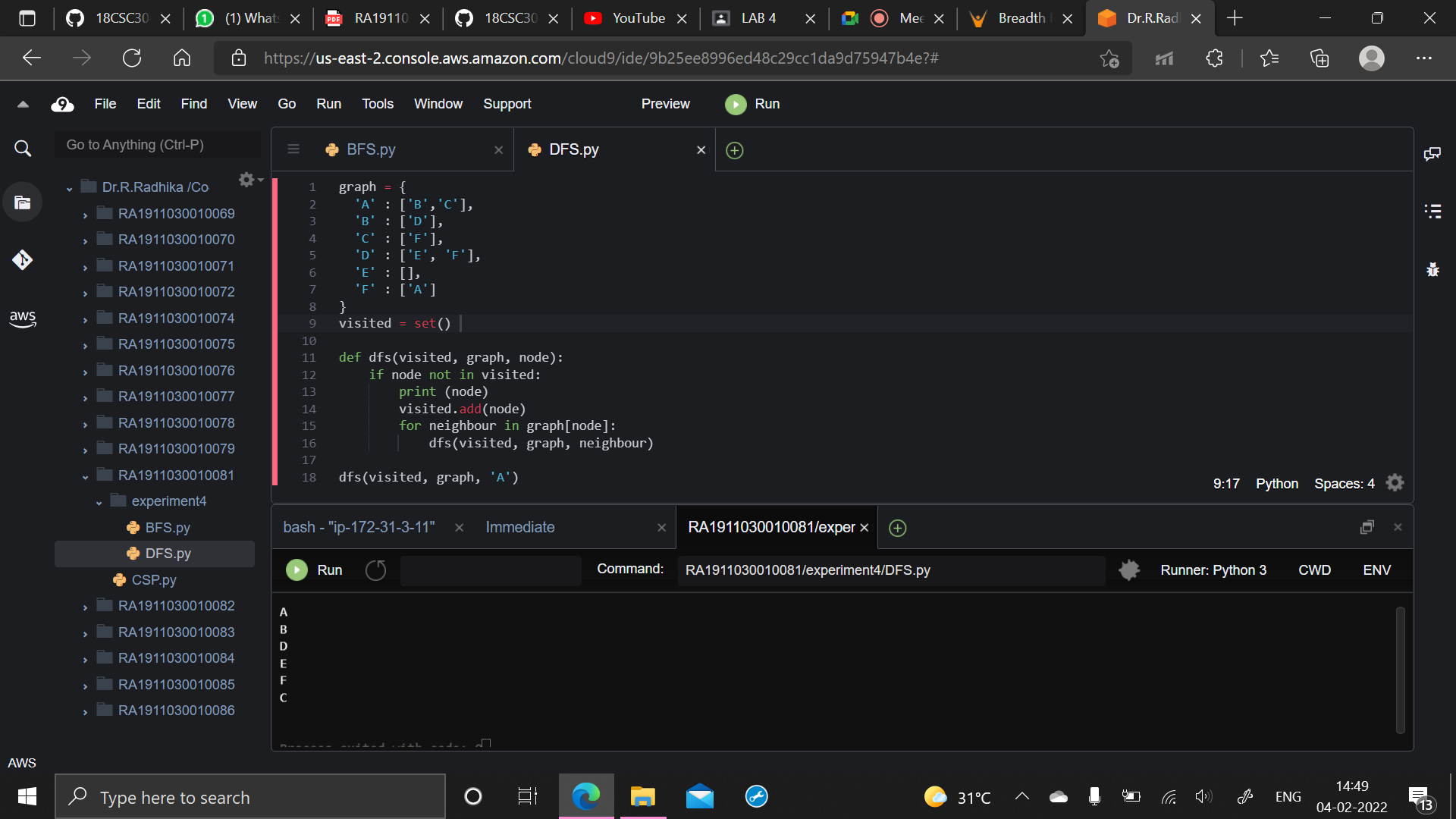
print (node)

visited**.**add(node)

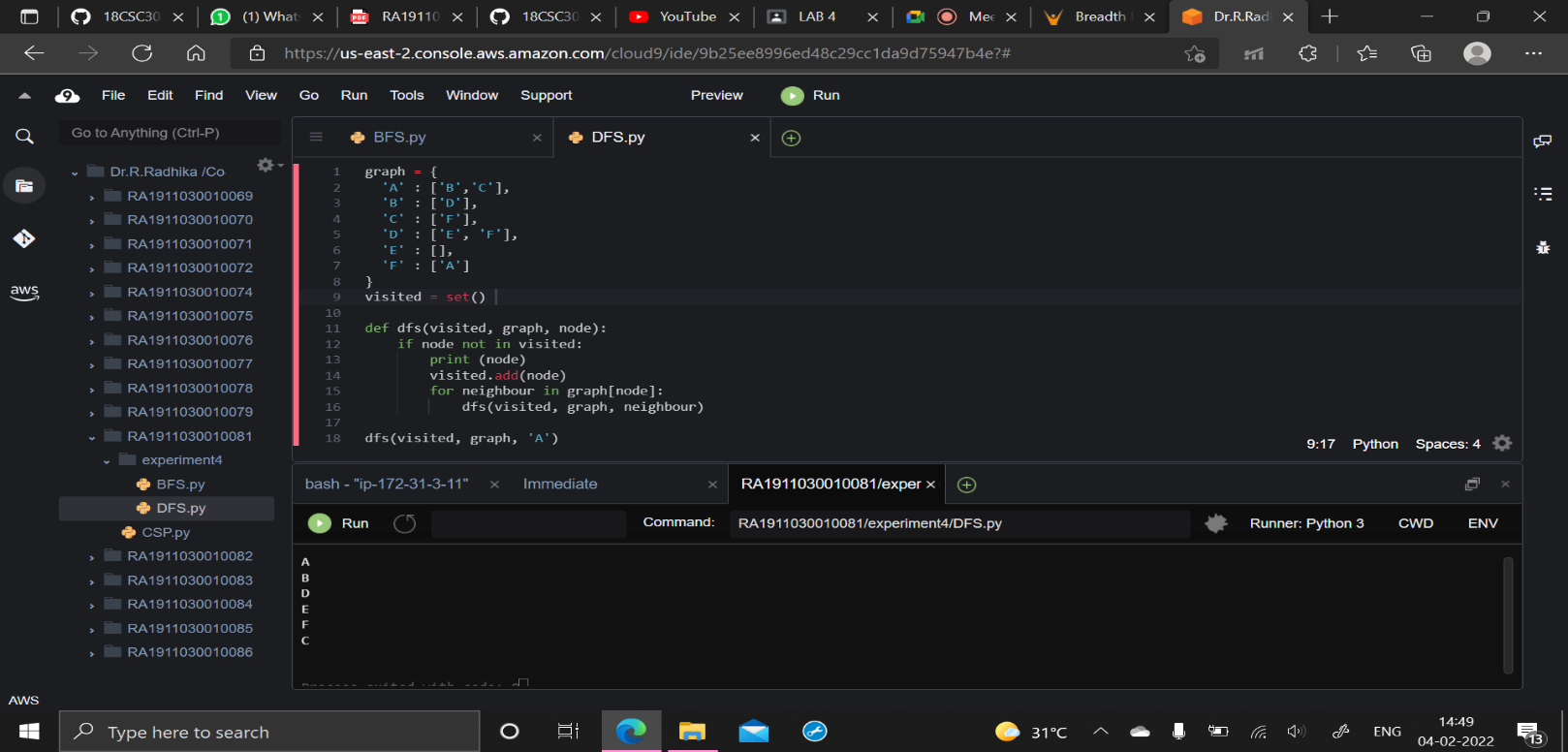
**for** neighbour **in** graph[node]:

dfs(visited, graph, neighbour)

dfs(visited, graph, 'A')



OUTPUT:



CODE BFS:

graph **=** {

'A' : ['B','C'],

'B' : ['D'],

'C' : ['F'],

'D' : ['E', 'F'],

'E' : [],

'F' : ['A']

}

visited **=** []

queue **=** []

**def** bfs(visited, graph, node):

visited**.**append(node)

queue**.**append(node)

**while** queue:

s **=** queue**.**pop(0)

print (s, end **=** " ")

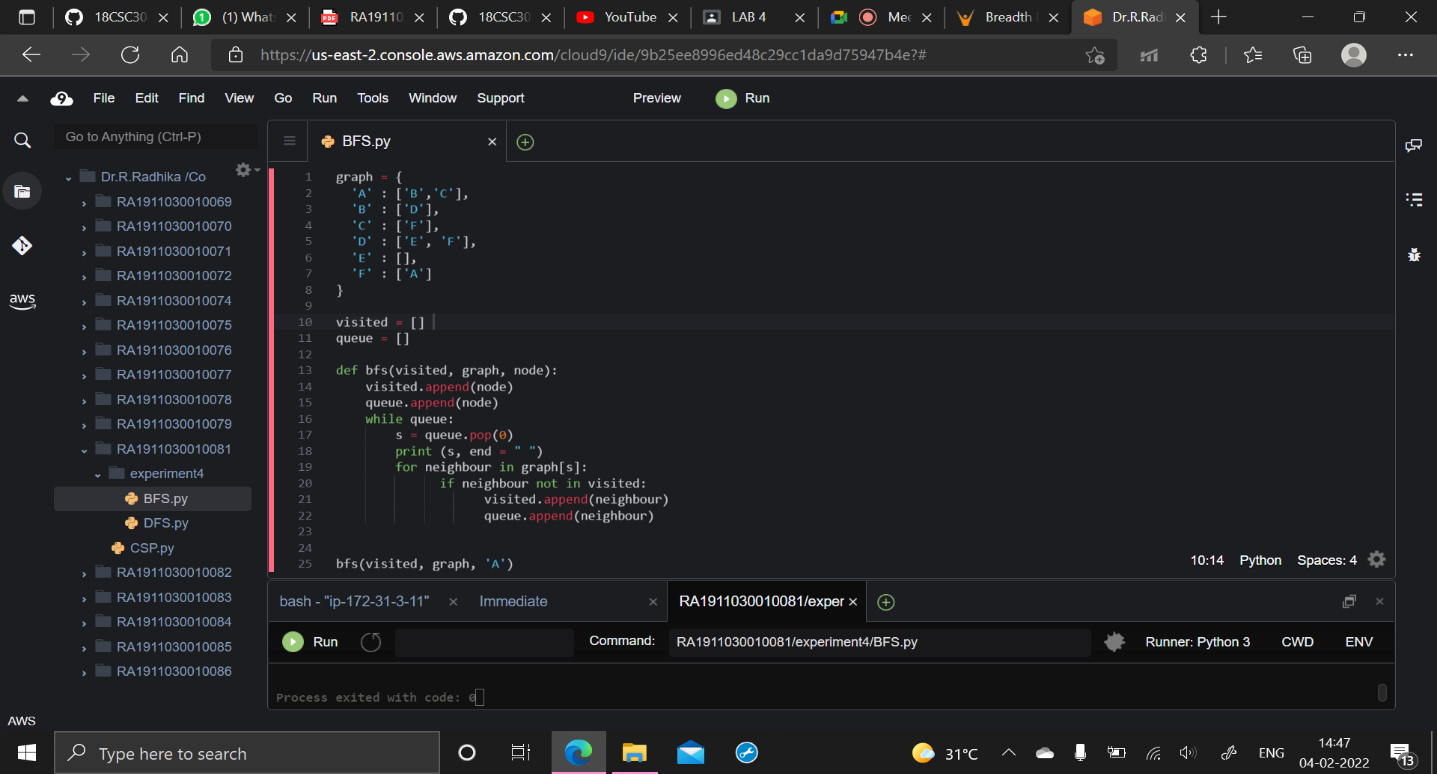
**for** neighbour **in** graph[s]:

**if** neighbour **not** **in** visited:

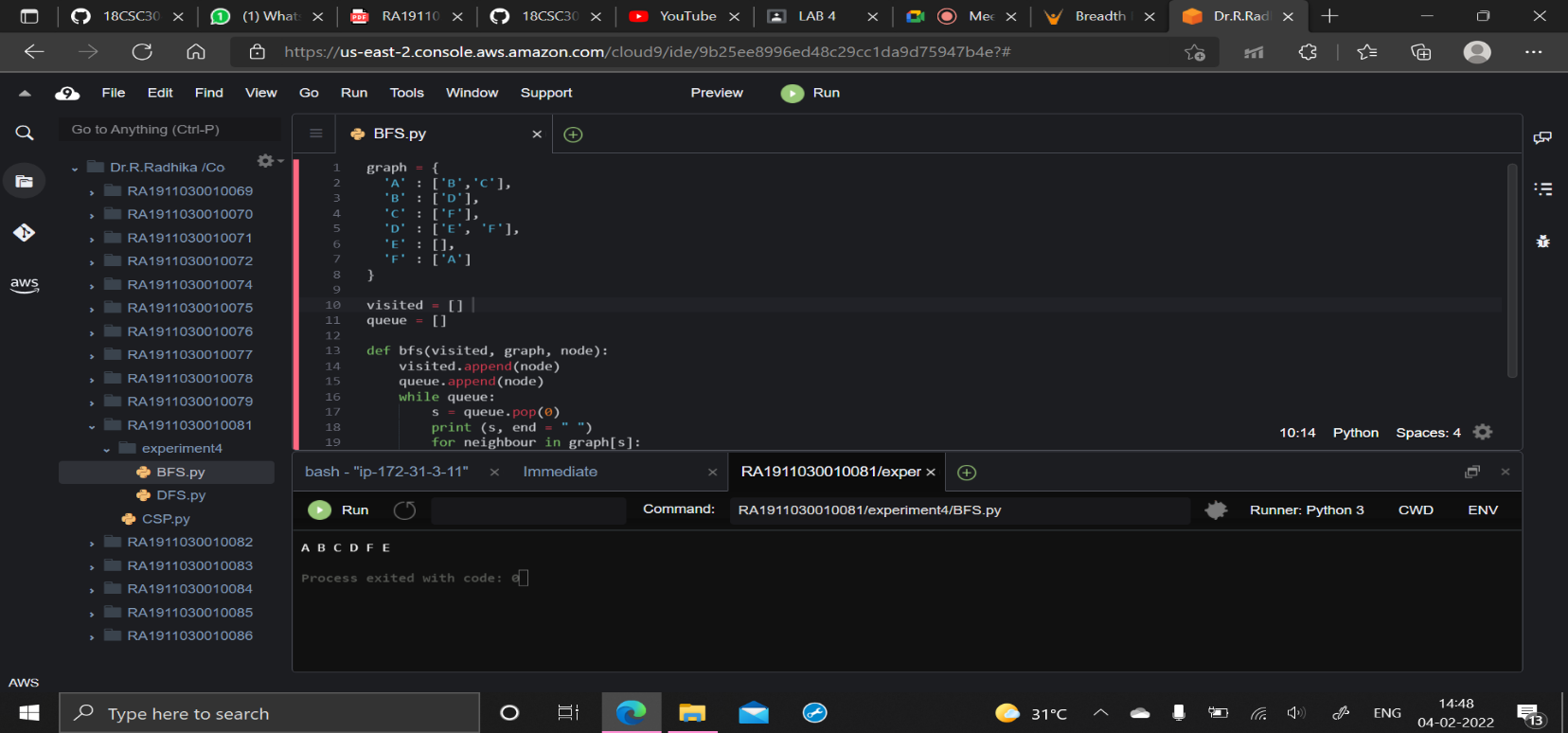
visited**.**append(neighbour)

queue**.**append(neighbour)

bfs(visited, graph, 'A')



OUTPUT:



RESULT:

Depth First Search and Breadth First Search was successfully implemented in Python.