BFS

graph = {

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

'B' : ['D', 'E'],

'C' : ['R'],

'D' : [],

'E' : ['R'],

'R' : []

}

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]:

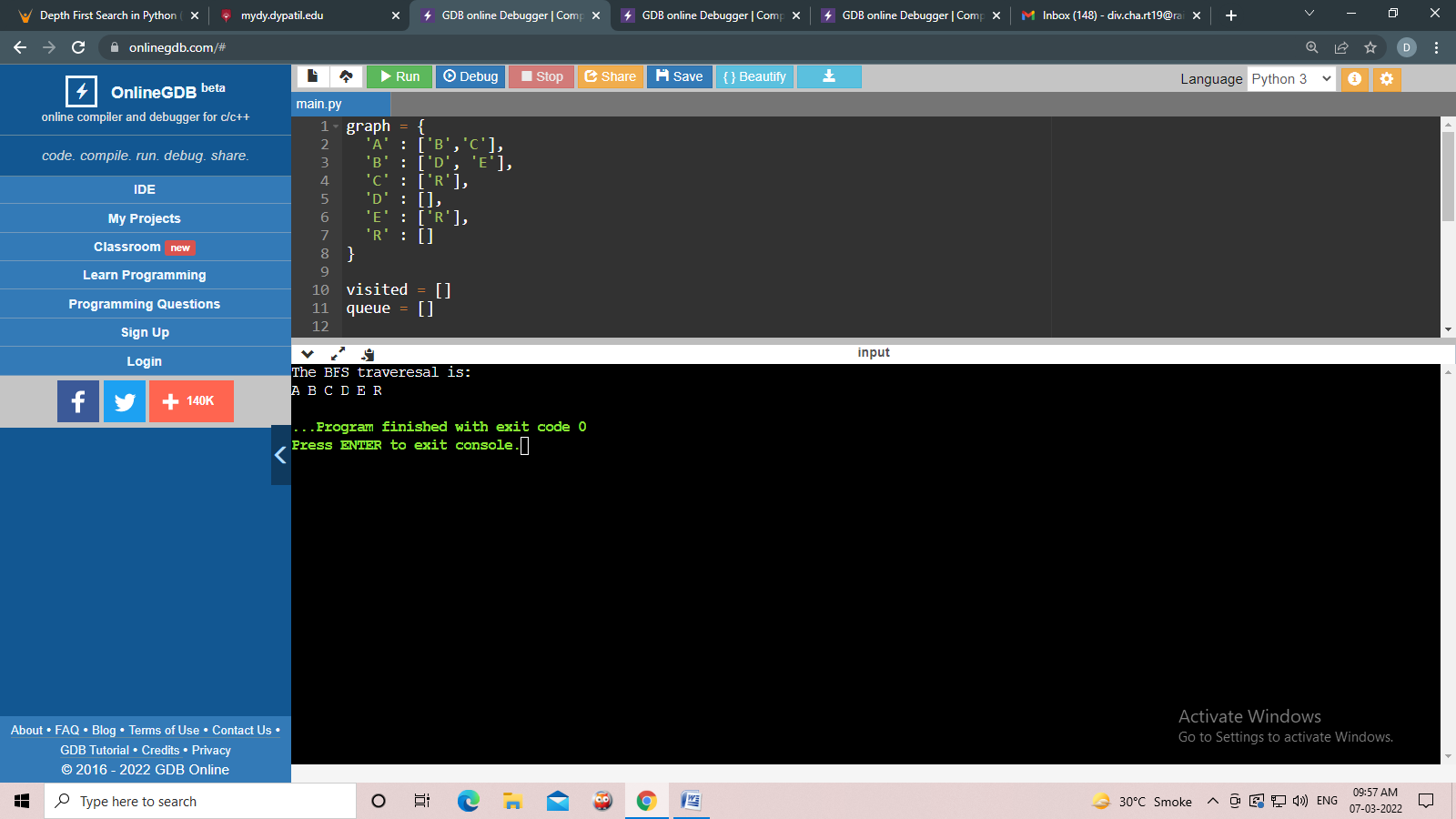
if neighbour not in visited:

visited.append(neighbour)

queue.append(neighbour)

print("The BFS traveresal is:")

bfs(visited, graph, 'A')



DFS

graph = {

'A' : ['B','D'],

'B' : ['K', 'C'],

'D' : ['T'],

'K' : [],

'C' : ['T'],

'T' : []

}

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)

print("The Depth-First Search is:")

dfs(visited, graph,'A')

