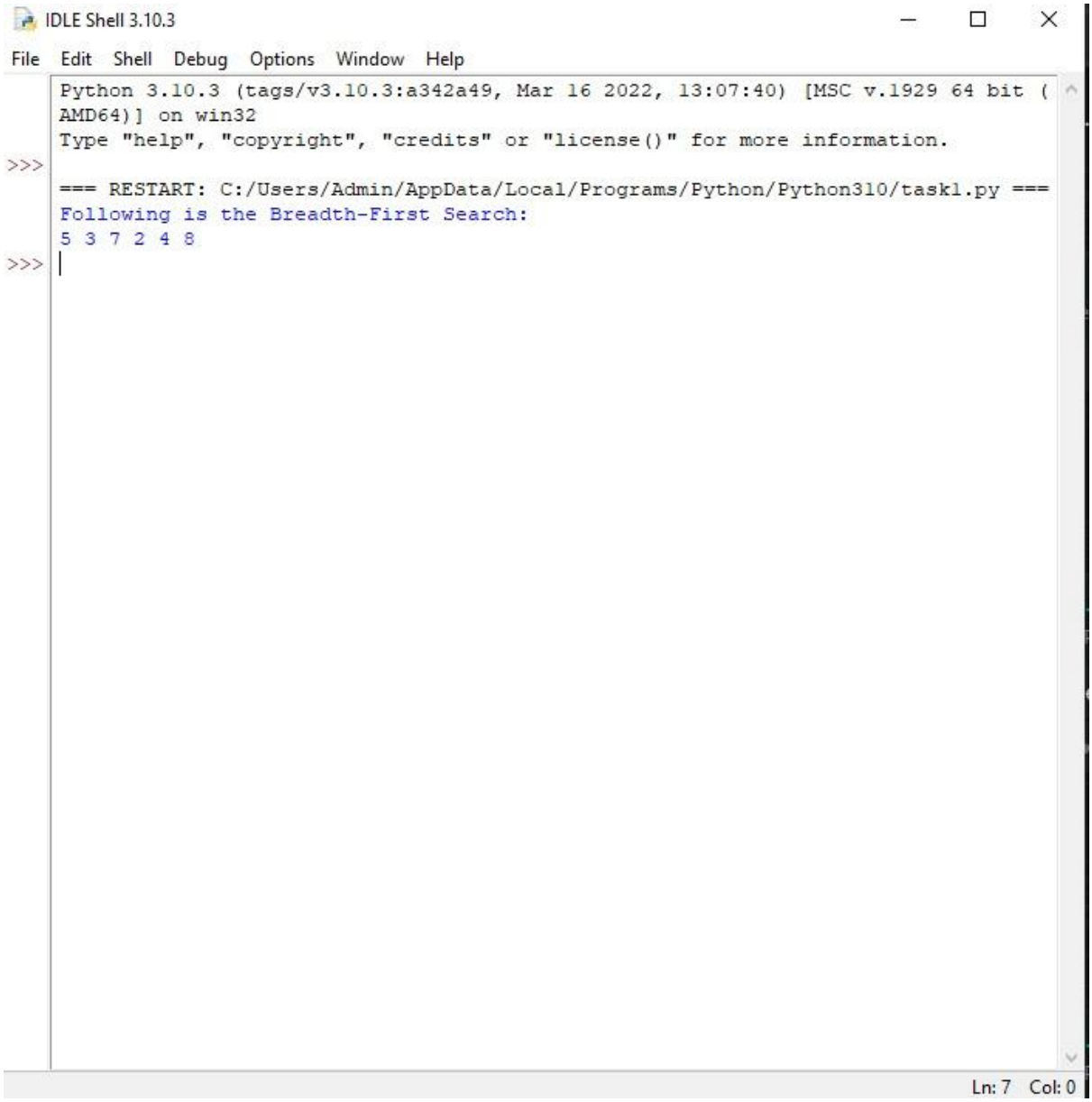
TASK 1A:

```
graph = {
  '5': ['3', '7'],
  '3': ['2', '4'],
  '7': ['8'],
  '2': [],
  '4': ['8'],
  '8': []
}

visited = []  # List for visited nodes.
queue = []    # Initialize a queue

def bfs(visited, graph, node):  # Function for BFS
  visited.append(node)
  queue.append(node)

  while queue:  # Creating loop to visit each node
    m = queue.pop(0)
    print(m, end=" ")

    for neighbour in graph[m]:
      if neighbour not in visited:
        visited.append(neighbour)
        queue.append(neighbour)

# Driver Code
print("Following is the Breadth-First Search:")
bfs(visited, graph, '5')  # Function calling
```

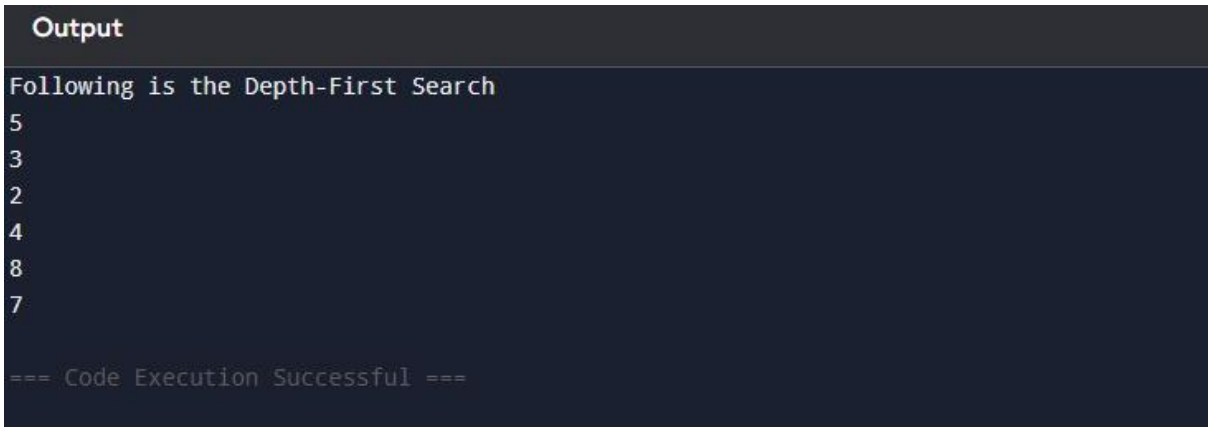
OUTPUT:


IDLE Shell 3.10.3
File Edit Shell Debug Options Window Help
Python 3.10.3 (tags/v3.10.3:a342a49, Mar 16 2022, 13:07:40) [MSC v.1929 64 bit (
AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
=== RESTART: C:/Users/Admin/AppData/Local/Programs/Python/Python310/task1.py ===
Following is the Breadth-First Search:
5 3 7 2 4 8
>>>
Ln: 7 Col: 0

TASK 1B:

```
graph = {
  '5': ['3', '7'],
  '3': ['2', '4'],
  '7': ['8'],
  '2': [],
  '4': ['8'],
  '8': []
}

visited = set()  # Set to keep track of visited nodes of graph.

def dfs(visited, graph, node):  # Function for DFS
  if node not in visited:
    print(node)
    visited.add(node)
    for neighbour in graph[node]:
      dfs(visited, graph, neighbour)

# Driver Code
print("Following is the Depth-First Search")
dfs(visited, graph, '5')
```

OUTPUT:


Output
Following is the Depth-First Search
5
3
2
4
8
7
=== Code Execution Successful ===