Name Swapnil De Sarkar Student ID: 231003004031

Assignment 2

Code:

```
# 1. Create the tree
class Node:
  def __init__(self, value):
    self.value = value
    self.children = []

# 2. Display the tree structure
def display_tree(node, level=0):
  print('\t' * level + str(node.value))
  for child in node.children:
    display_tree(child, level + 1)
# 3. Perform depth-first traversal
def depth_first_traversal(node):
  print(node.value, end=' ')
  for child in node.children:
    depth_first_traversal(child)
# 4. Search for a node using depth-first search
def dfs_search(root, target):
  stack = [root]
  visited = []
  found = False
  while stack:
    node = stack.pop()
    visited.append(node.value)
    if node.value == target:
      print(f"Found {target}: Traversal order {visited}")
      found = True
      break

    for child in reversed(node.children):
      stack.append(child)
  if not found:
    print(f"Nodes visited: {visited}")
    print(f"{target} NOT FOUND")
```

Name Swapnil De Sarkar Student ID: 231003004031

```
# 1.Build the tree
root = Node(1)
root.children = [Node(2), Node(3), Node(4)]

node2 = root.children[0]
node2.children = [Node(5), Node(6)]

node4 = root.children[2]
node4.children = [Node(7), Node(8), Node(9)]

node9 = node4.children[2]
node9.children = [Node(10), Node(11), Node(12)]

# 2. Display the tree structure
print("Tree Structure:")
display_tree(root)

# 3. Perform depth-first traversal
print("\nDepth First Traversal:")
depth_first_traversal(root)
print()

# 4. Search for nodes 8, 10, and 13
print("\nSearching for 8:")
dfs_search(root, 8)

print("\nSearching for 10:")
dfs_search(root, 10)

print("\nSearching for 13:")
dfs_search(root, 13)
```

Name Swapnil De Sarkar Student ID: 231003004031

Output:

Tree Structure:

1

2

5

6

3

4

7

8

9

10

11

12

Depth First Traversal:

1 2 5 6 3 4 7 8 9 10 11 12

Searching for 8:

Found 8: Traversal order [1, 2, 5, 6, 3, 4, 7, 8]

Searching for 10:

Found 10: Traversal order [1, 2, 5, 6, 3, 4, 7, 8, 9, 10]

Searching for 13:

Nodes visited: [1, 2, 5, 6, 3, 4, 7, 8, 9, 10, 11, 12]

13 NOT FOUND