Name: Swapnil De Sarkar Student ID: 231003004031

Assignment 3

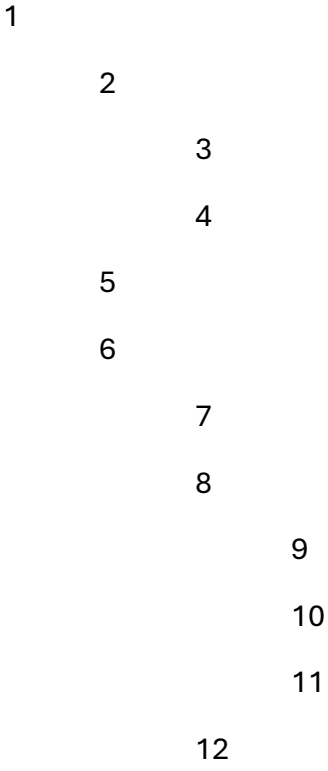
```
# 1. Create the tree
class Node:
  def __init__(self, value):
    self.value = value
    self.children = []

# 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 breadth-first traversal
def breadth_first_traversal(root):
  queue = [root]
  while queue:
    node = queue.pop(0)
    print(node.value, end=' ')
    queue.extend(node.children)
  print()

# 4. Search for a node using breadth-first search
def bfs_search(root, target):
  queue = [root]
  visited = []
  found = False
  while queue:
    node = queue.pop(0)
    visited.append(node.value)
    if node.value == target:
      print(f"Found {target}: Traversal order {visited}")
      found = True
      break
```

```
        queue.extend(node.children)
    if not found:
        print(f"Nodes visited: {visited}")
        print(f"{target} NOT FOUND")

# 1. Build the tree
root = Node(1)
root.children = [Node(2), Node(5), Node(6)]

node2 = root.children[0]
node2.children = [Node(3), Node(4)]

node6 = root.children[2]
node6.children = [Node(7), Node(8), Node(12)]

node8 = node6.children[1]
node8.children = [Node(9), Node(10), Node(11)]

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

# 3. Perform breadth-first traversal
print("\nBreadth First Traversal:")
breadth_first_traversal(root)

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

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

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

Name: Swapnil De Sarkar Student ID: 231003004031

Output:

Tree Structure:



Breadth First Traversal:

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

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, 12, 9, 10]

Searching for 13:

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

13 NOT FOUND