Esha Asif
034
BSAI-3A
Task 5
AI LAB

Task 1

```
#def dfs(start):
#     stack=[start]
#     visit=set()

#     while stack:
#         node=stack.pop()
#         if node not in visit:
#             print(node.value,end=" ")
#             visit.add(node)
#             stack.extend(neighbor for neighbor in node.neighbors if neighbor not in visit)

# class node:
#     def __init__(self, value):
#         self.value =value
#         self.neighbors =[]

# node1 = node(1)
# node2 = node(2)
# node3 = node(3)
# node4 = node(4)
# node5 = node(5)
# node1.neighbors = [node2, node3]
# node2.neighbors = [node4, node5]
# node3.neighbors = [node5]
# dfs(node1)
```

```
1 3 5 2 4
```

Task2:

```
# class Node:
```

```
#     def __init__(self, value):
#         self.value = value
#         self.left = None
#         self.right = None
# def preorder(node):
#     if node:
#         print(node.value, end=" ")
#         preorder(node.left)
#         preorder(node.right)

# def inorder(node):
#     if node:
#         inorder(node.left)
#         print(node.value, end=" ")
#         inorder(node.right)

# def postorder(node):
#     if node:
#         postorder(node.left)
#         postorder(node.right)
#         print(node.value, end=" ")

# root = Node(1)
# root.left = Node(2)
# root.right = Node(3)
# root.left.left = Node(4)
# root.left.right = Node(5)
# root.right.left = Node(6)
# root.right.right = Node(7)

# print("preorder traversal")
# preorder(root)
# print("inorder traversal")
# inorder(root)
# print("postorder traversal")
# postorder(root)
```

```
preorder traversal
1 2 4 5 3 6 7 /ninorder traversal
4 2 5 1 6 3 7 postorder traversal
4 5 2 6 7 3 1
```