TERM PROJECT #2

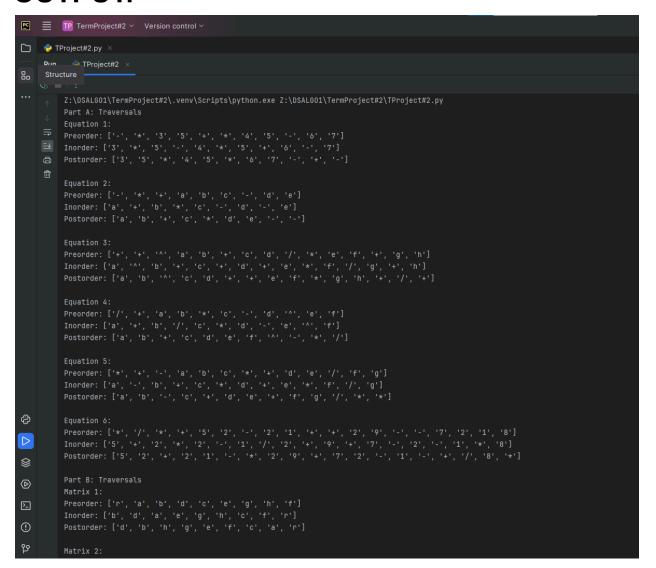
CODE:

```
TProject#2.py ×
    Structure s Node:
           def __init__(self, value):
    self.value = value
         def preorder_traversal(root):
           return [root.value] + preorder_traversal(root.left) + preorder_traversal(root.right)
         def postorder_traversal(root):
(
         def construct_expression_tree(expression):
           operators = set(['+', '-', '*', '/', '^'])
寥
(D)
             for char in expression.split():
               if char not in operators:
2
①
                    node = Node(char)
අ
```

```
▼ TP TermProject#2 ∨ Version control ∨
TProject#2.py ×
                  else:
80
                      stack.append(node)
              return stack[-1] if stack else None
      45 ∨ def construct_tree_from_matrix(vertices, left_children, right_children):
             for parent, left, right in zip(vertices, left_children, right_children):
                     nodes[parent].right = nodes[right]
            return nodes[vertices[0]]
      58 v expressions = [
6
      trees_part_a = [construct_expression_tree(expr) for expr in expressions]
♦
Ð
      70 ∨ for i, tree in enumerate(trees_part_a, 1):
2
              print("Preorder:", preorder_traversal(tree))
print("Inorder:", inorder_traversal(tree))
①
```

```
🥏 TProject#2.py 🗵
           for i, tree in enumerate(trees_part_a, 1):
80
           vertices_list = [
           left_children_list = [
            ["a", "b", "-", "e", "-", "-", "-", "-", "-"],
["a", "c", "-", "-", "-", "-", "-"],
["a", "-", "d", "f", "-", "-", "-"],
           right_children_list = [
      99 trees_part_b = [
                for vertices, left, right in zip(vertices_list, left_children_list, right_children_list)
@
寥
(
\bullet
①
```

OUTPUT:



```
Matrix 2:
Preorder: ['r', 'a', 'c', 'd', 'b', 'e', 'f', 'g']
Inorder: ['c', 'a', 'd', 'r', 'b', 'e', 'f', 'g']
Postorder: ['c', 'd', 'a', 'g', 'f', 'e', 'b', 'r']

Matrix 3:
Preorder: ['r', 'a', 'c', 'f', 'b', 'd', 'e']
Inorder: ['a', 'f', 'c', 'r', 'd', 'b', 'e']
Postorder: ['f', 'c', 'a', 'd', 'e', 'b', 'r']

Matrix 4:
Preorder: ['r', 'a', 'c', 'g', 'h', 'd', 'b', 'e', 'i', 'f']
Inorder: ['g', 'c', 'h', 'a', 'd', 'r', 'i', 'e', 'b', 'f']
Postorder: ['g', 'h', 'c', 'd', 'a', 'i', 'e', 'f', 'b', 'r']
```