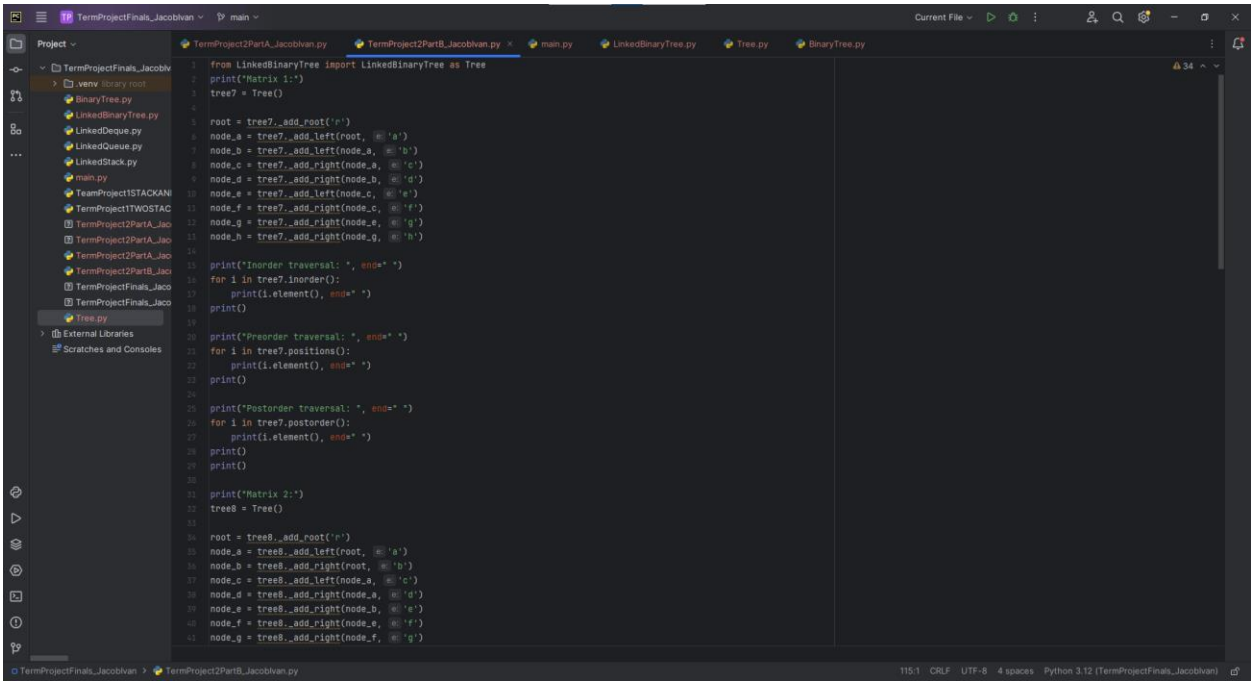
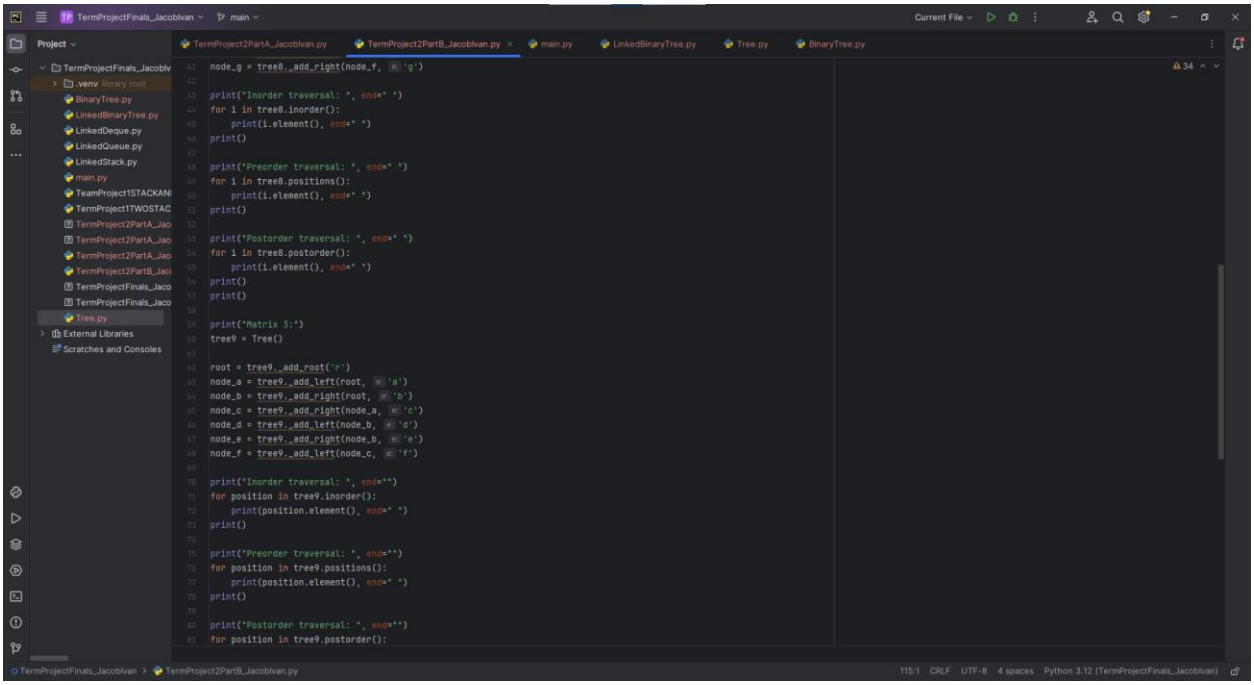
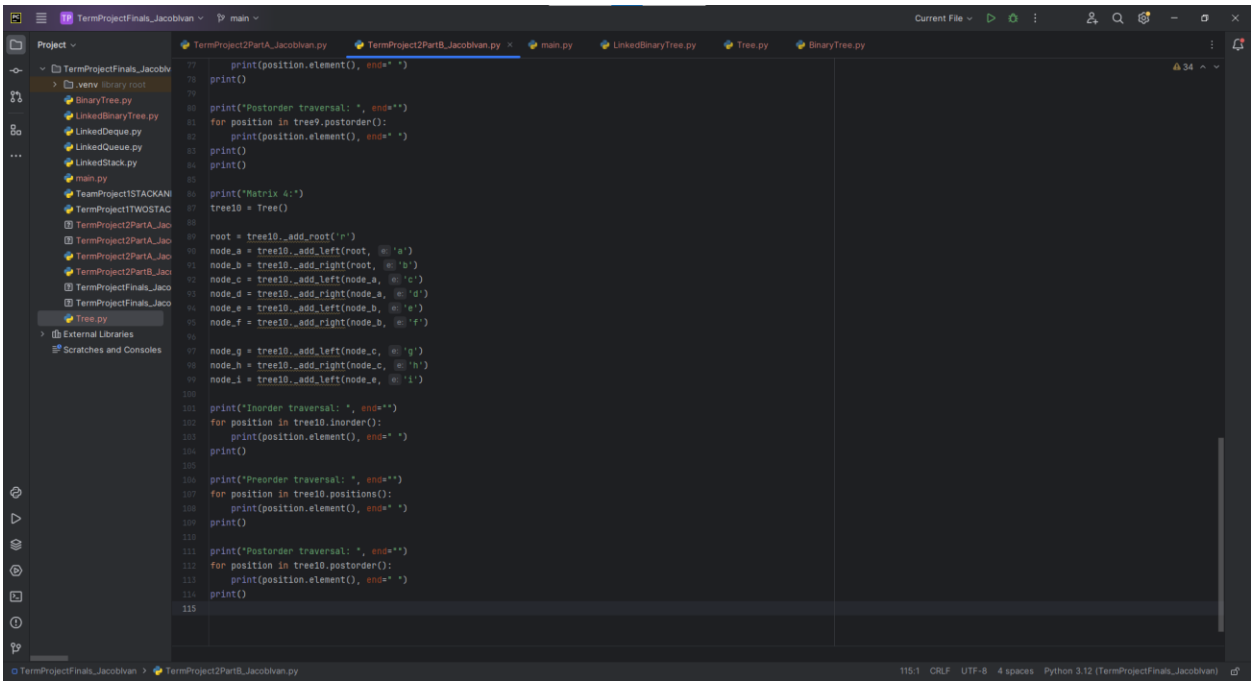
Code:



```
from LinkedBinaryTree import LinkedBinaryTree as Tree
print("Matrix 1:")
tree7 = Tree()

root = tree7._add_root('r')
node_a = tree7._add_left(root, e='a')
node_b = tree7._add_left(node_a, e='b')
node_c = tree7._add_right(node_a, e='c')
node_d = tree7._add_right(node_b, e='d')
node_e = tree7._add_left(node_c, e='e')
node_f = tree7._add_right(node_c, e='f')
node_g = tree7._add_right(node_e, e='g')
node_h = tree7._add_right(node_g, e='h')

print("Inorder traversal: ", end=" ")
for i in tree7.inorder():
    print(i.element(), end=" ")
print()

print("Preorder traversal: ", end=" ")
for i in tree7.positions():
    print(i.element(), end=" ")
print()

print("Postorder traversal: ", end=" ")
for i in tree7.postorder():
    print(i.element(), end=" ")
print()
print()

print("Matrix 2:")
tree8 = Tree()

root = tree8._add_root('r')
node_a = tree8._add_left(root, e='a')
node_b = tree8._add_right(root, e='b')
node_c = tree8._add_left(node_a, e='c')
node_d = tree8._add_right(node_a, e='d')
node_e = tree8._add_right(node_b, e='e')
node_f = tree8._add_right(node_e, e='f')
node_g = tree8._add_right(node_f, e='g')
```

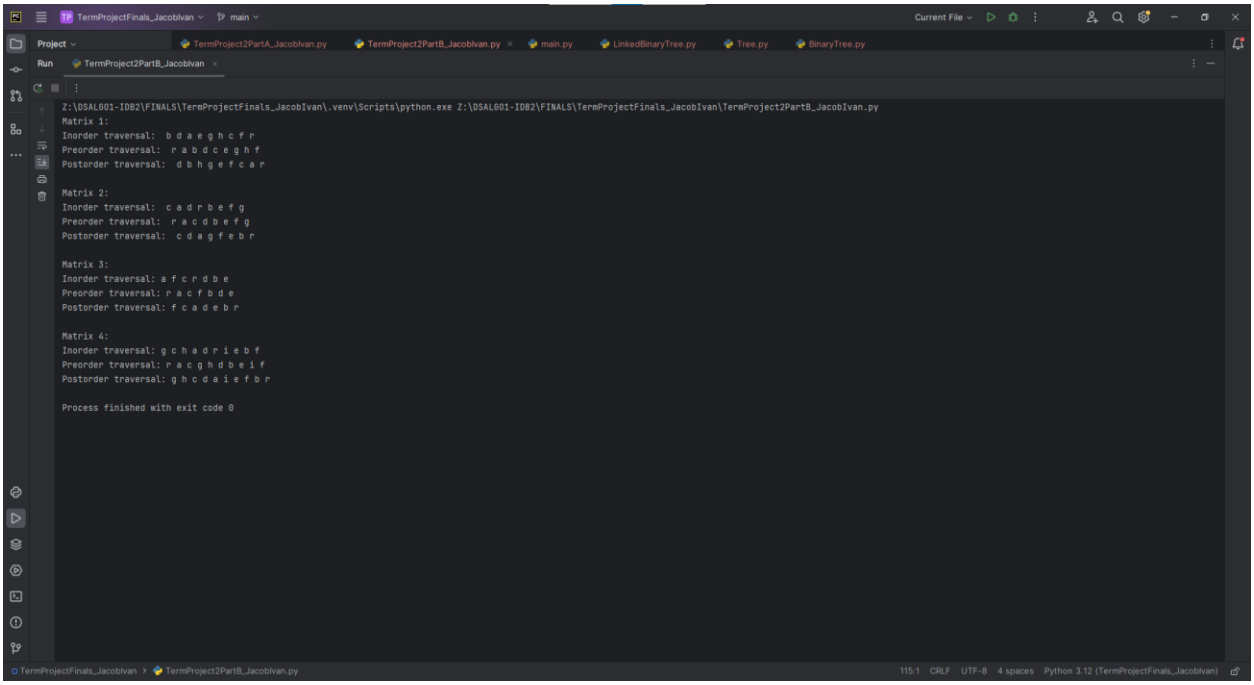


```
node_g = tree8._add_right(node_f, e='g')

print("Inorder traversal: ", end=" ")
for i in tree8.inorder():
    print(i.element(), end=" ")
print()

print("Preorder traversal: ", end=" ")
for i in tree8.positions():
    print(i.element(), end=" ")
print()

print("Postorder traversal: ", end=" ")
for i in tree8.postorder():
    print(i.element(), end=" ")
print()
print()

print("Matrix 3:")
tree9 = Tree()

root = tree9._add_root('r')
node_a = tree9._add_left(root, e='a')
node_b = tree9._add_right(root, e='b')
node_c = tree9._add_right(node_a, e='c')
node_d = tree9._add_left(node_b, e='d')
node_e = tree9._add_right(node_b, e='e')
node_f = tree9._add_left(node_c, e='f')

print("Inorder traversal: ", end="")
for position in tree9.inorder():
    print(position.element(), end=" ")
print()

print("Preorder traversal: ", end="")
for position in tree9.positions():
    print(position.element(), end=" ")
print()

print("Postorder traversal: ", end="")
for position in tree9.postorder():
```



```
    print(position.element(), end=" ")
print()

print("Postorder traversal: ", end="")
for position in tree9.postorder():
    print(position.element(), end=" ")
print()
print()

print("Matrix 4:")
tree10 = Tree()

root = tree10._add_root('r')
node_a = tree10._add_left(root, e='a')
node_b = tree10._add_right(root, e='b')
node_c = tree10._add_left(node_a, e='c')
node_d = tree10._add_right(node_a, e='d')
node_e = tree10._add_left(node_b, e='e')
node_f = tree10._add_right(node_b, e='f')

node_g = tree10._add_left(node_c, e='g')
node_h = tree10._add_right(node_c, e='h')
node_i = tree10._add_left(node_e, e='i')

print("Inorder traversal: ", end="")
for position in tree10.inorder():
    print(position.element(), end=" ")
print()

print("Preorder traversal: ", end="")
for position in tree10.positions():
    print(position.element(), end=" ")
print()

print("Postorder traversal: ", end="")
for position in tree10.postorder():
    print(position.element(), end=" ")
print()
```

OUTPUT:



```
Z:\DSALG01-ID82\FINALS\TermProjectFinals_JacobIvan\.venv\Scripts\python.exe Z:\DSALG01-ID82\FINALS\TermProjectFinals_JacobIvan\TermProject2PartB_JacobIvan.py
Matrix 1:
Inorder traversal:  b d a e g h c f r
Preorder traversal:  r a b d c e g h f
Postorder traversal:  d b h g e f c a r

Matrix 2:
Inorder traversal:  c a d r b e f g
Preorder traversal:  r a c d b e f g
Postorder traversal:  c d a g f e b r

Matrix 3:
Inorder traversal: a f c r d b e
Preorder traversal: r a c f b d e
Postorder traversal: f c a d e b r

Matrix 4:
Inorder traversal: g c h a d r i e b f
Preorder traversal: r a c g h d b e i f
Postorder traversal: g h c d a i e f b r

Process finished with exit code 0
```