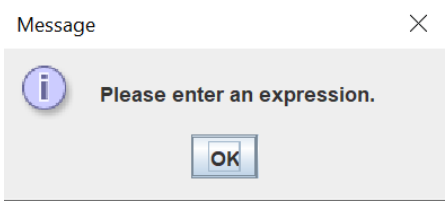
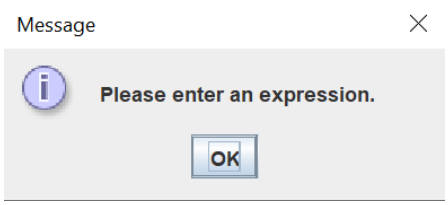
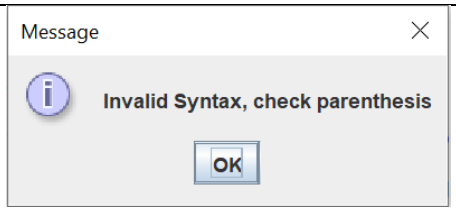
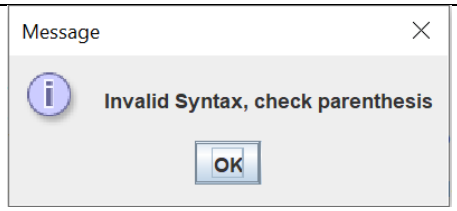
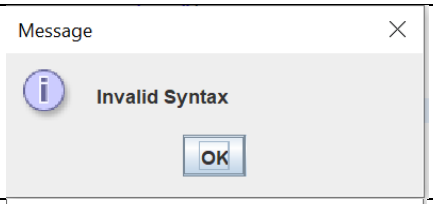
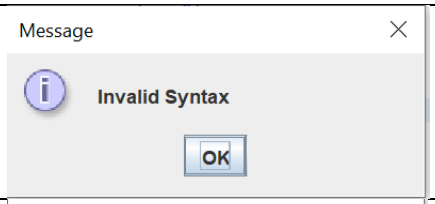
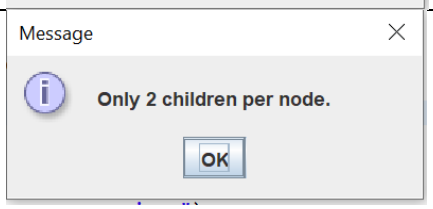
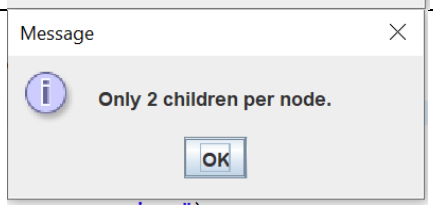
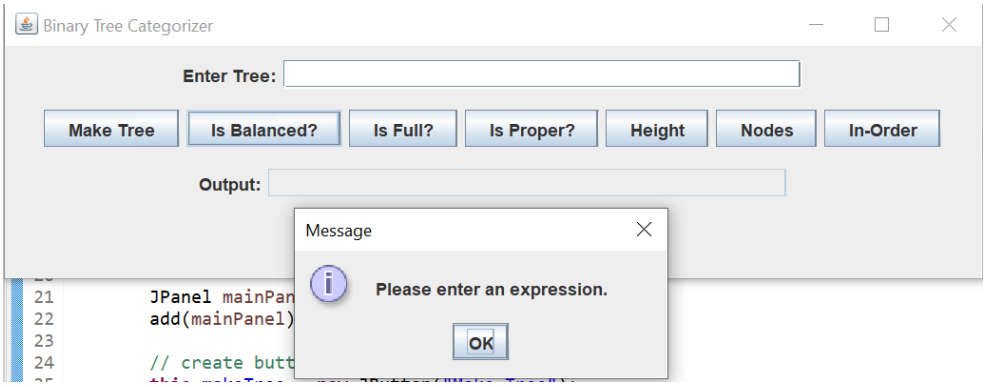
Test Case	Input	Expected Output	Actual Output	Pass?
1	*click any of the buttons with no entry*			Yes
2	(A1(G(j)(1))(z(5))) *Click “Make Tree*			Yes
3	A(G(j)(1))(z(5))) *Click “Make Tree*			Yes
4	(A(G(j)(1)(z(5))) *Click “Make Tree*			Yes
5	(A(G(j)(1))(z(5))) *Click all buttons*	See output below	See output below	Yes
6	(d(c(a)(b))(g(e)(f))) *Click all buttons*	See output below	See output below	Yes
7	(s(a)(z)) *Click all buttons*	See output below	See output below	Yes
8	(1(2(3))) *Click all buttons*	See output below	See output below	Yes
9	(a(s(a))(z(g(h)))) *Click all buttons*	See output below	See output below	Yes

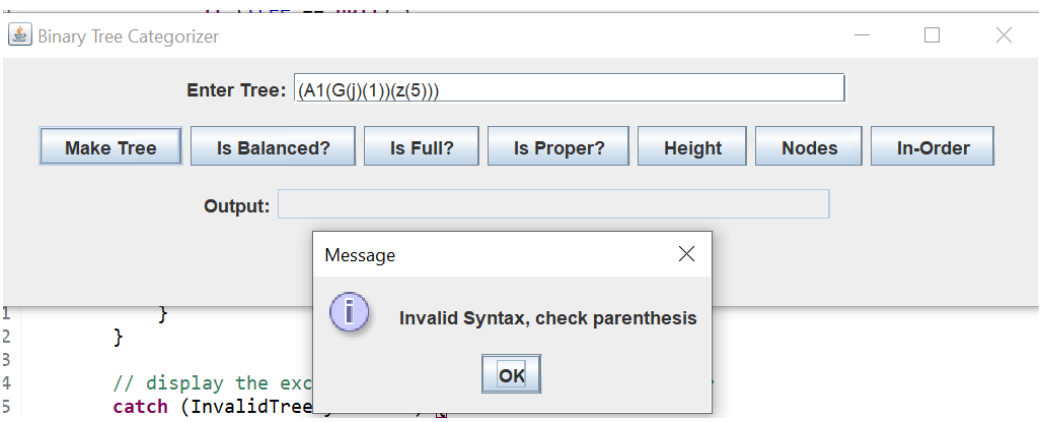
Test Case 1 Output

Test Case 1 displays the error handling of the program. If not entry is entered by the user, a JOptionPane is display informing the user that they need to enter an expression.



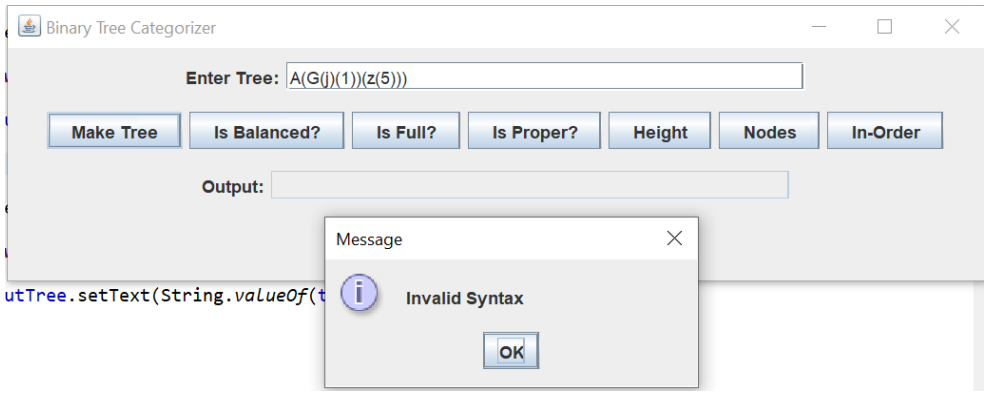
Test Case 2 Output

Test Case 2 displays the error if the user does not enter a valid expression. The "A1" is the issue. If the 1 is removed, the program will allow the tree to be created.



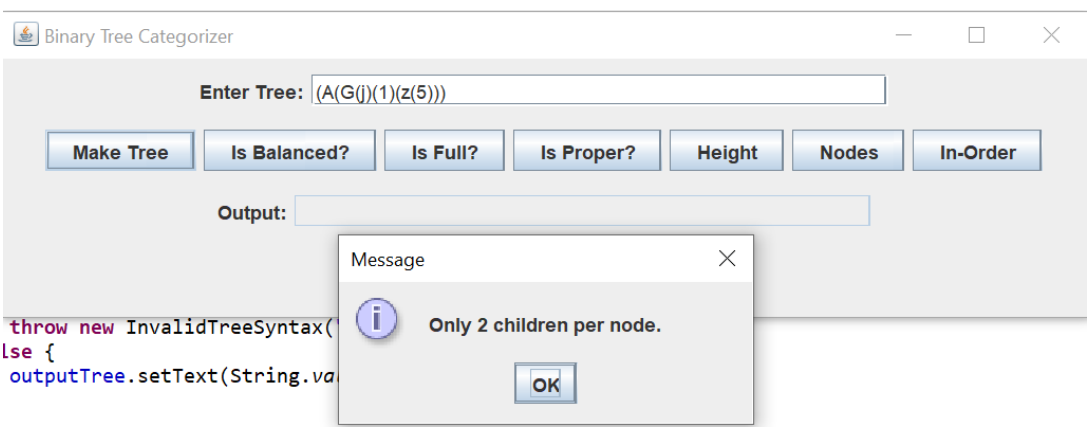
Test Case 3 Output

Test Case 3 displays another error where the user does not enter a valid expression. The issue in this case is there is a missing parenthesis at the front of the expression.

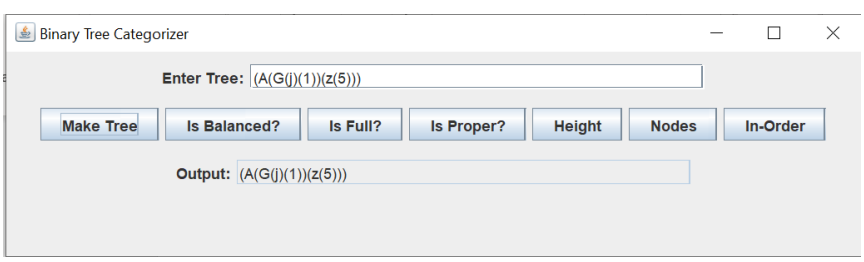


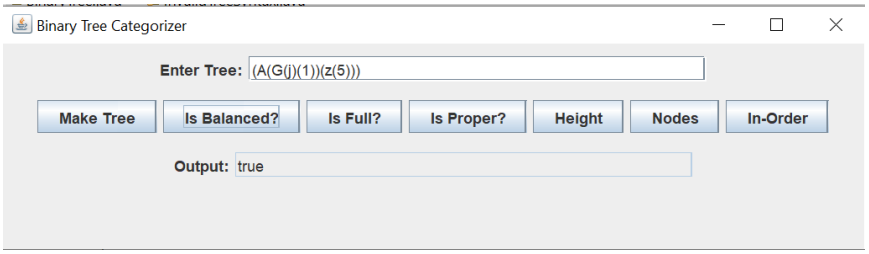
Test Case 4 Output

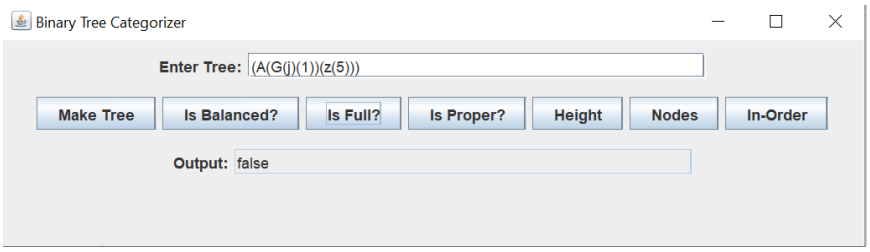
Test Case 4 displays another error handling where there are too many children. There is a missing closing parenthesis after the "1" which causes the program to think that the expression wants to make Z and 5 children of G as well.

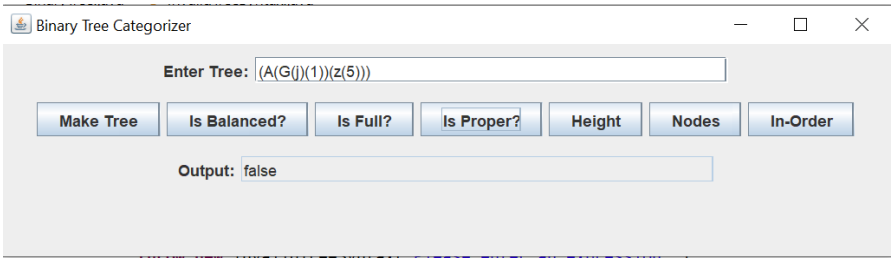


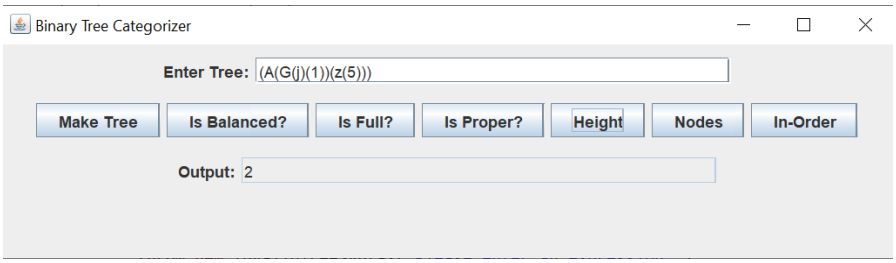
Test Case 5 Output

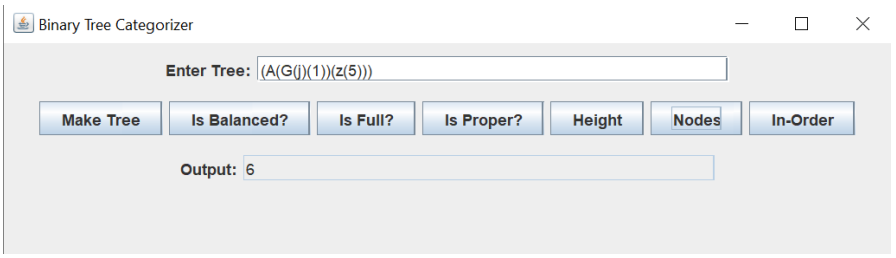


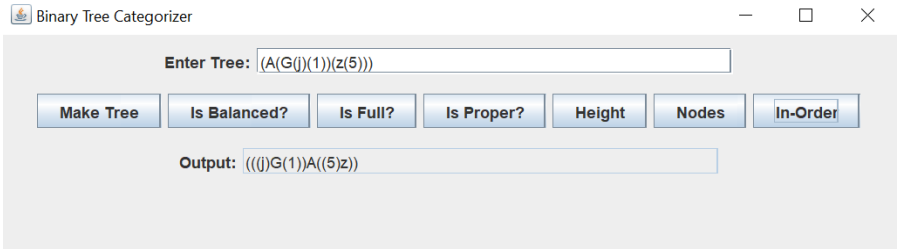

Binary Tree Categorizer
Enter Tree: (A(G(j)(1))(z(5)))
Make Tree
Is Balanced?
Is Full?
Is Proper?
Height
Nodes
In-Order
Output: true


Binary Tree Categorizer
Enter Tree: (A(G(j)(1))(z(5)))
Make Tree
Is Balanced?
Is Full?
Is Proper?
Height
Nodes
In-Order
Output: false

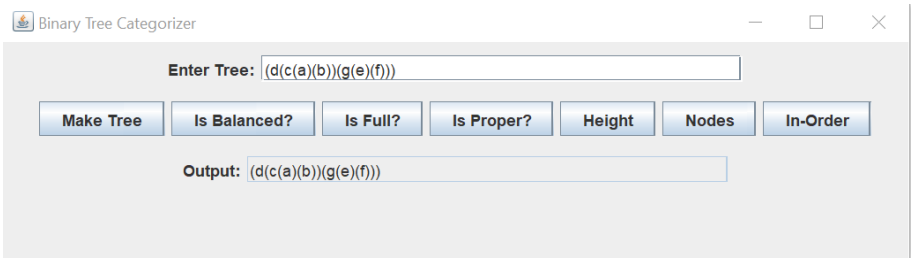

Binary Tree Categorizer
Enter Tree: (A(G(j)(1))(z(5)))
Make Tree
Is Balanced?
Is Full?
Is Proper?
Height
Nodes
In-Order
Output: false


Binary Tree Categorizer
Enter Tree: (A(G(j)(1))(z(5)))
Make Tree
Is Balanced?
Is Full?
Is Proper?
Height
Nodes
In-Order
Output: 2


Binary Tree Categorizer
Enter Tree: (A(G(j)(1))(z(5)))
Make Tree
Is Balanced?
Is Full?
Is Proper?
Height
Nodes
In-Order
Output: 6

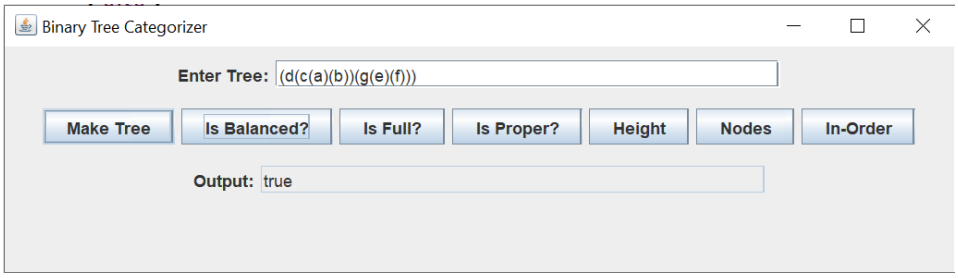

Binary Tree Categorizer
Enter Tree: (A(G(j)(1))(z(5)))
Make Tree
Is Balanced?
Is Full?
Is Proper?
Height
Nodes
In-Order
Output: (((j)G(1))A((5)z))

Test Case 6 Output



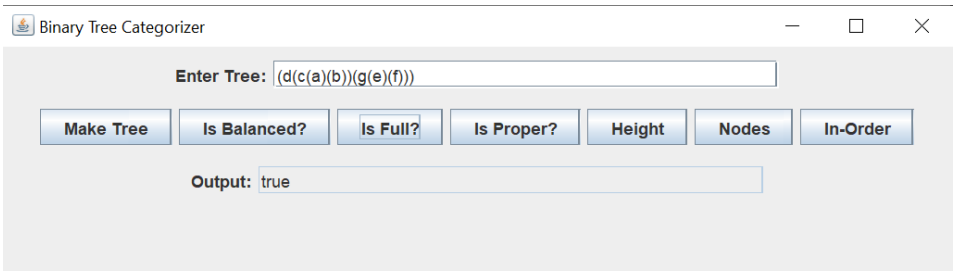
Enter Tree: (d(c(a)(b))(g(e)(f)))

Output: (d(c(a)(b))(g(e)(f)))



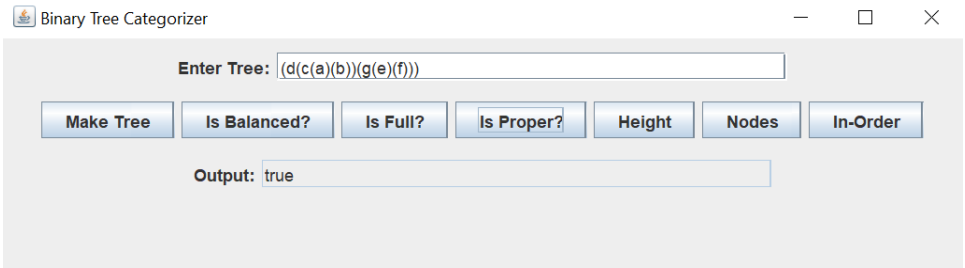
Enter Tree: (d(c(a)(b))(g(e)(f)))

Output: true



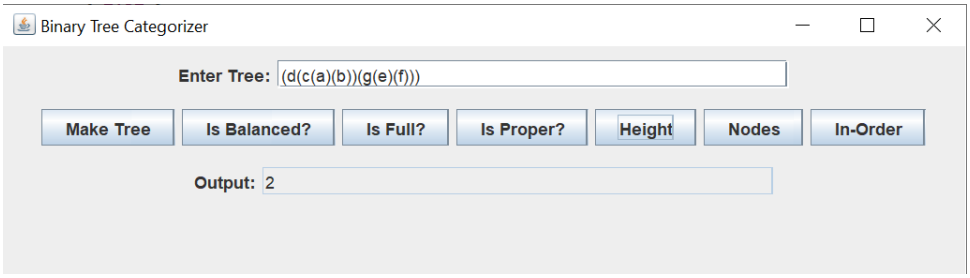
Enter Tree: (d(c(a)(b))(g(e)(f)))

Output: true



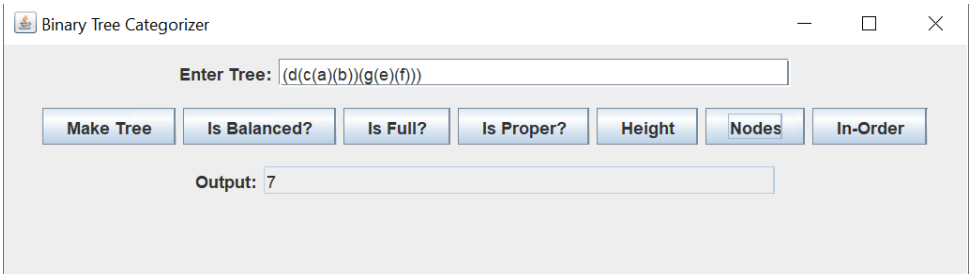
Enter Tree: (d(c(a)(b))(g(e)(f)))

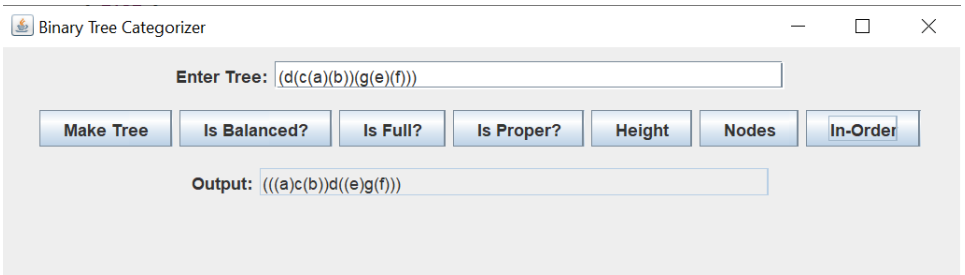
Output: true



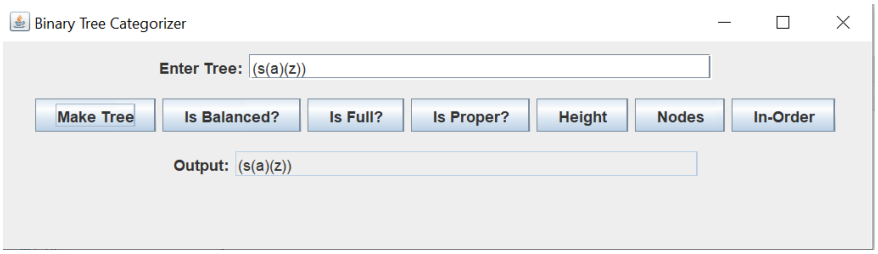
Enter Tree: (d(c(a)(b))(g(e)(f)))

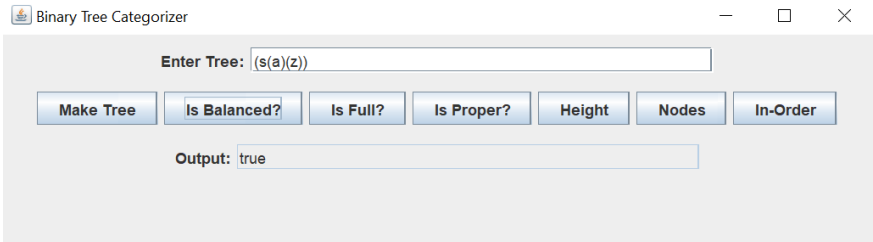
Output: 2

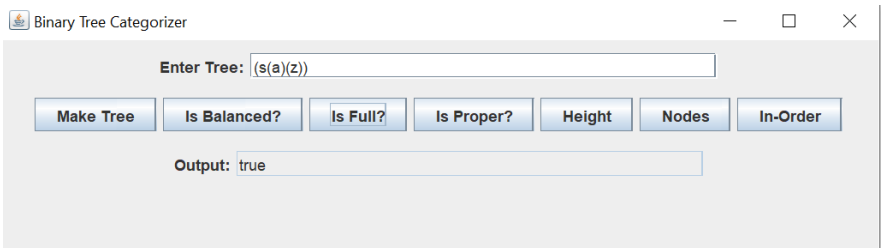


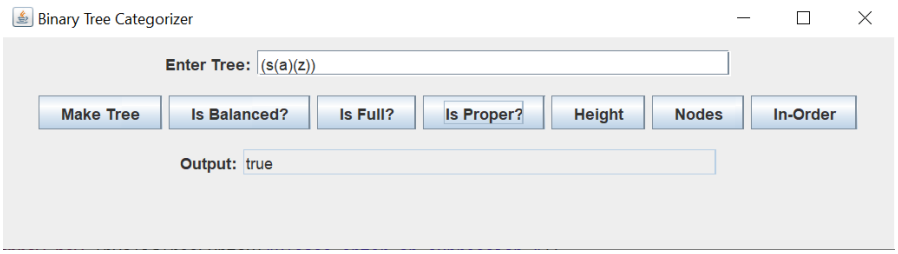


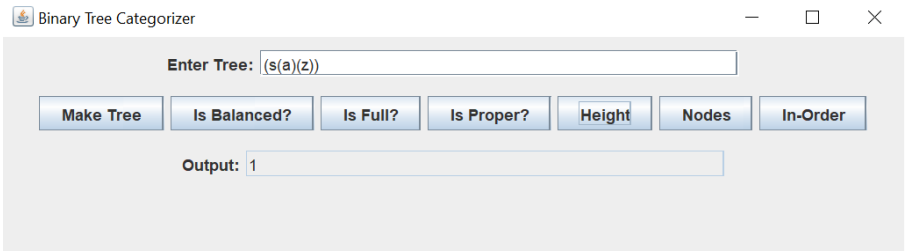
Test Case 7 Output

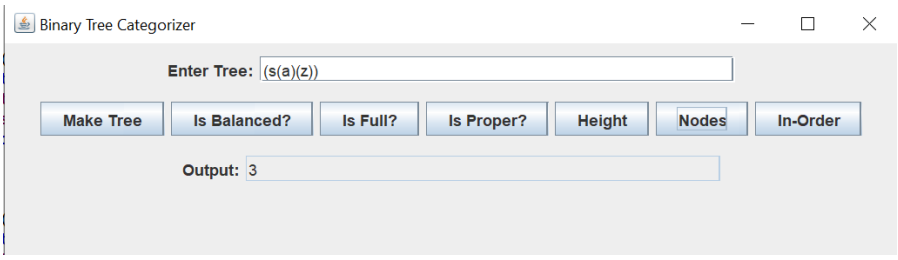


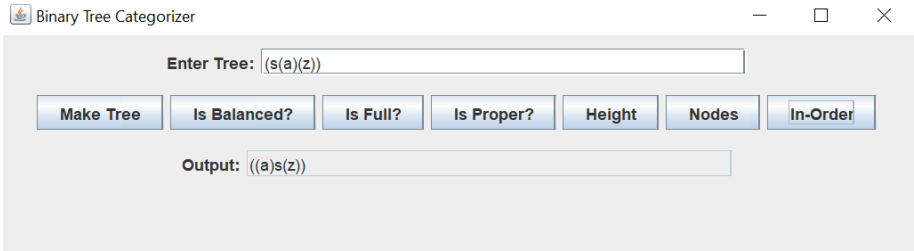




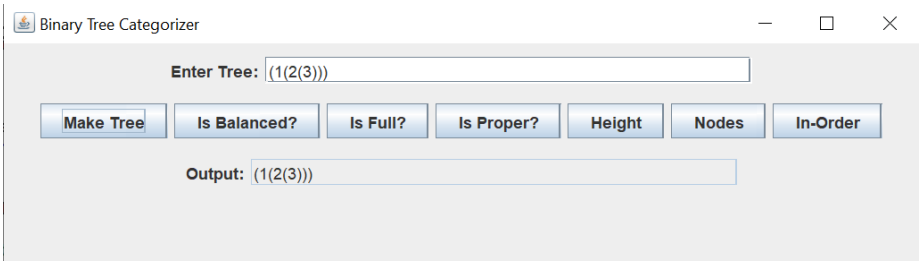


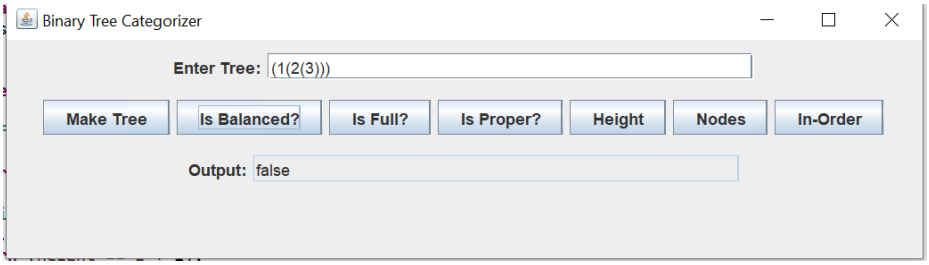






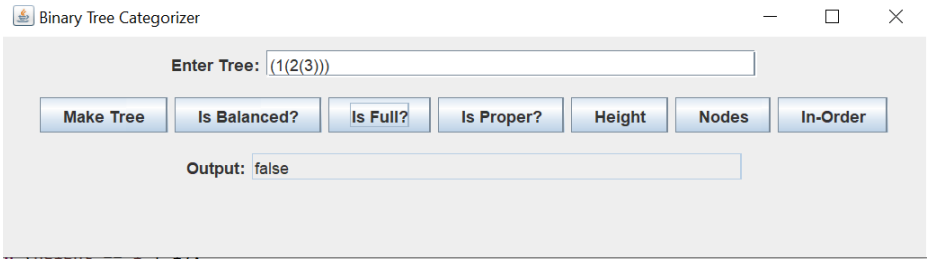
Test Case 8 Output





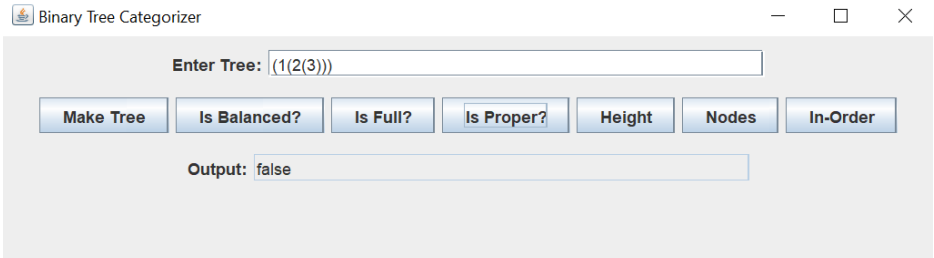
Enter Tree: (1(2(3)))

Output: false



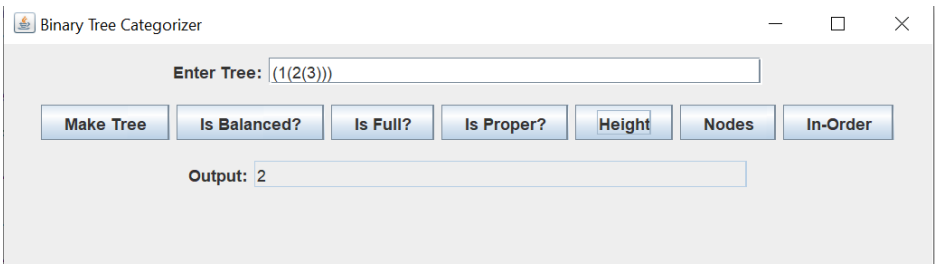
Enter Tree: (1(2(3)))

Output: false



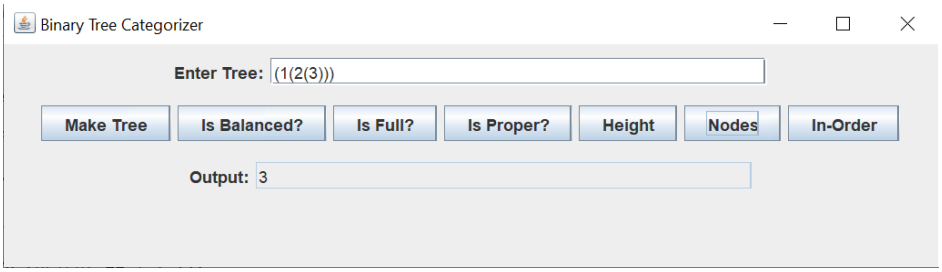
Enter Tree: (1(2(3)))

Output: false



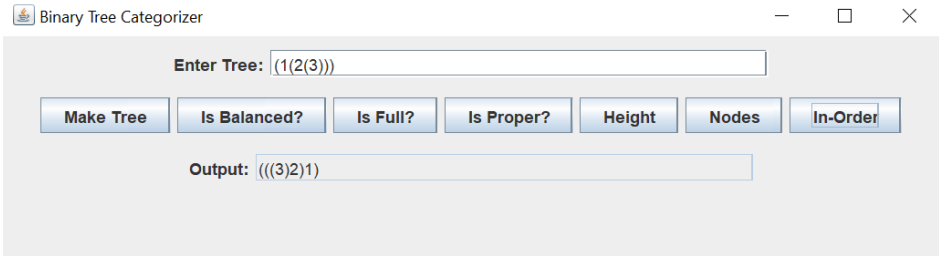
Enter Tree: (1(2(3)))

Output: 2



Enter Tree: (1(2(3)))

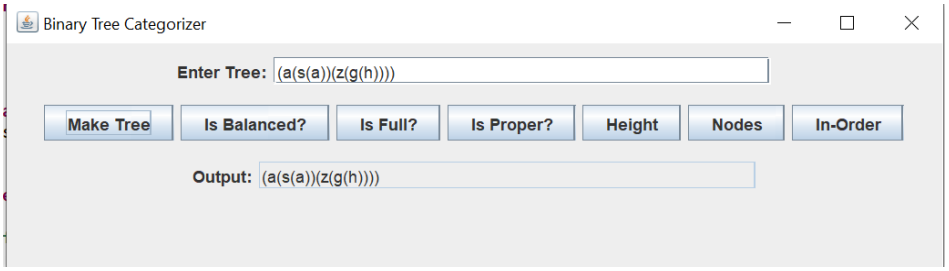
Output: 3



Enter Tree: (1(2(3)))

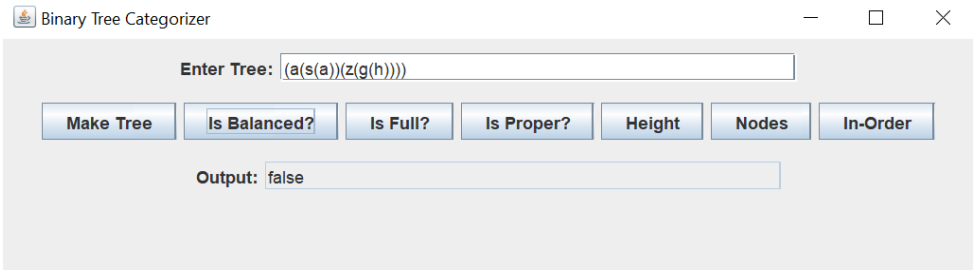
Output: (((3)2)1)

Test Case 9 Output



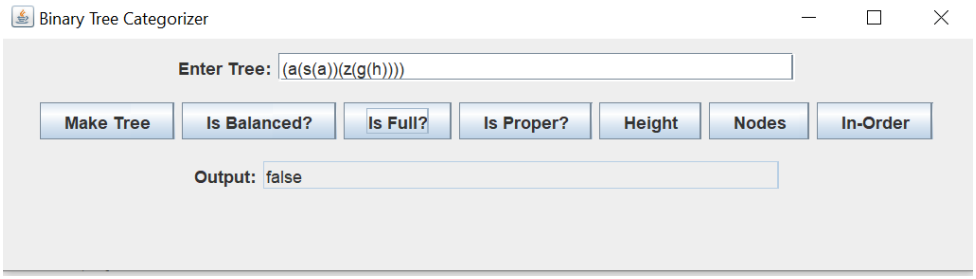
Enter Tree: (a(s(a))(z(g(h))))

Output: (a(s(a))(z(g(h))))



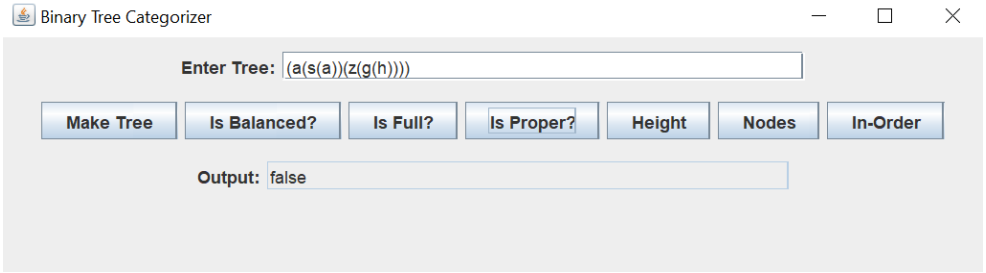
Enter Tree: (a(s(a))(z(g(h))))

Output: false



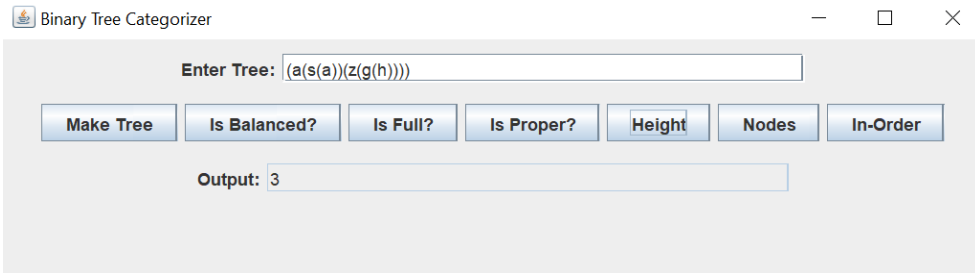
Enter Tree: (a(s(a))(z(g(h))))

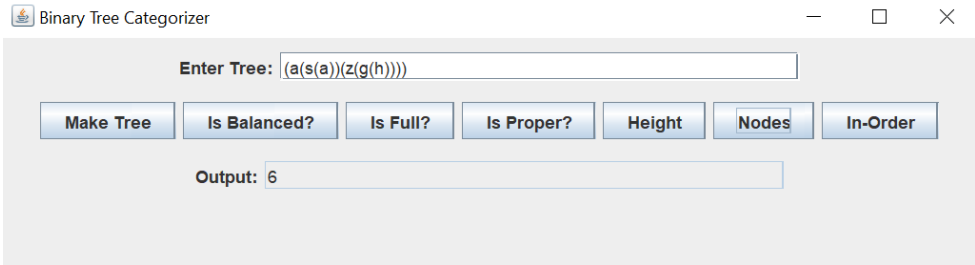
Output: false

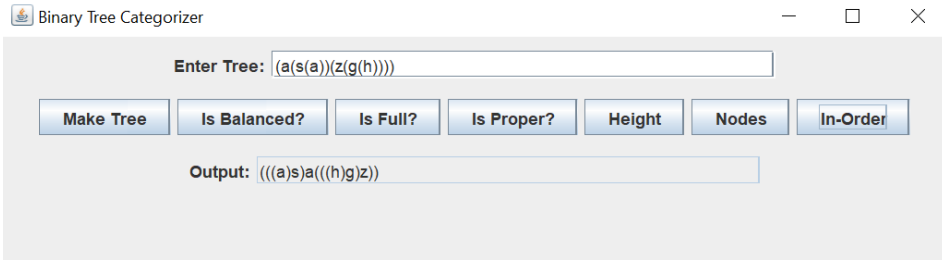


Enter Tree: (a(s(a))(z(g(h))))

Output: false


Binary Tree Categorizer
Enter Tree: (a(s(a))(z(g(h))))
Make Tree
Is Balanced?
Is Full?
Is Proper?
Height
Nodes
In-Order
Output: 3


Binary Tree Categorizer
Enter Tree: (a(s(a))(z(g(h))))
Make Tree
Is Balanced?
Is Full?
Is Proper?
Height
Nodes
In-Order
Output: 6


Binary Tree Categorizer
Enter Tree: (a(s(a))(z(g(h))))
Make Tree
Is Balanced?
Is Full?
Is Proper?
Height
Nodes
In-Order
Output: (((a)s)a(((h)g)z))