1. FIRST TREE

INORDER = 32, 30, 40, 55
PRE-ORDER = 40, 30, 32, 55
POSTORDER = 32, 30, 55, 40

SECOND TREE

INORDER = 32, 30, 35, 40, 50, 55
PRE-ORDER = 40, 30, 32, 35, 50, 55
POSTORDER = 32, 35, 30, 50, 55, 40

2. FIRST TREE

INORDER = 4, 5, 6, 2, 3, 5, 0, 1, 3, 8, 9, 10, 9, 4, 12, 1, 7, 11.

PREORDER = 7, 1, 0, 3, 2, 5, 4, 6, 11, 9, 8, 10, 12.

POSTORDER = 4, 6, 5, 2, 3, 0, 1, 8, 10, 9, 12, 4, 7.

SECOND TREE

INORDER = 0, 1, 2, 3, 4, 5, 6, 3.

PREORDER = 3, 1, 0, 2, 5, 4, 6.

POSTORDER = 0, 2, 1, 4, 6, 5, 3

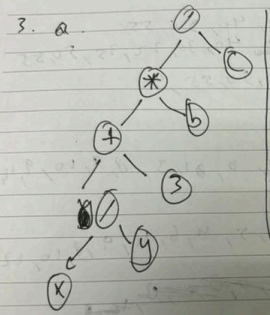
THIRD TREE

INORDER = 0, 1, 2, 3, 4, 5

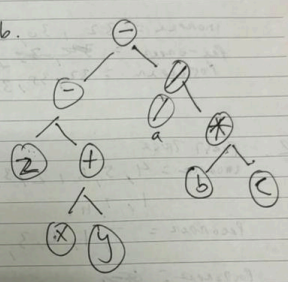
PREORDER = 3, 1, 0, 2, 5, 4

POSTORDER = 0, 2, 1, 4, 5, 3.

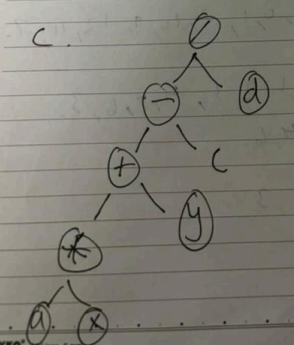
3. a.



b.

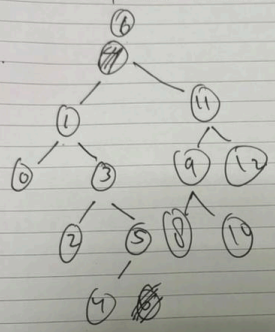


c.



4. The statement is supposed to ensure that the tree follows the convention expected in Java programming, which is particularly relevant when discussing tree traversal algorithms.

5.



Take 6 to replace 7