**Question 1 (34 Points):** Consider the tree given below and perform the following traversal, and print the nodes accordingly.

1. Inorder(1) /\* do the in-order traversal from 1\*/   
   **16,8,17,4,18,9,19,2,20,10,21,5,22,11,23,1,24,12,25,6,26,13,27,3,28,14,29,7,30,15,31**
2. II. preorder(1) /\* do the pre-order traversal from 1\*/

**1,2,4,8,16,17,18,19,5,10,20,21,11,22,23,3,6,12,24,25,13,26,27,7,14,28,29,15,30,31**

1. III. postorder(1) /\* do the post-order traversal from 1\*/

**16,17,8,18,19,9,4,20,21,10,22,23,11,5,2,24,25,12,26,27,13,6,28,29,14,30,31,15,7,3,1**

**Question 2 (33 Points)**: Consider the following equation:

𝑦 + { (𝑥 + 𝑦) ∗ 𝑧 (𝑦 ∗ 𝑧) − 𝑥 + (𝑥 ∗ 𝑥) + (𝑦 ∗ 𝑧) − 𝑥 𝑥 + 𝑦 + 𝑧 }

Now do the following:

1. Convert it to a binary expression tree.

Diagram

Description automatically generated

1. II. Do preorder traversal of this tree

**+ y + \* x y / z x - \* + y + z + x + y + z**

1. III. Do post-order traversal of this tree

**y z + + y z + + \* + x – x z x y z + \* + y +**

**Question 3 (23 Points):** Consider the following binary search tree (BST). Now, delete node 161, 150, 137 and 141. Then redraw the BST at each step.

Diagram

Description automatically generated

**Question 4 (10 Points):** You have a binary tree and a BST; both contain N nodes. If you search for a random node from these trees, in the worst case, how many iterations will be needed for the

1. binary tee? Justify your answer

**The worst case scenario for searching a binary tree is O(N) iterations. This is because the tree could be unbalanced and out of place, therefore making the worst case scenario O(N) iterations.**

1. (ii) BST? Justify your answer

**The worst case scenario for searching a BST is also O(N) iterations. This is also if the tree is unbalanced, so the worst case would be O(N) iterations.**