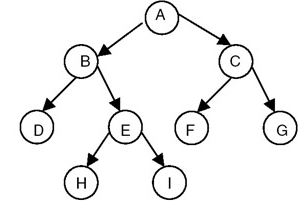
1. **Traverse the given tree using Preorder, Postorder and Backward Inorder traversals.**

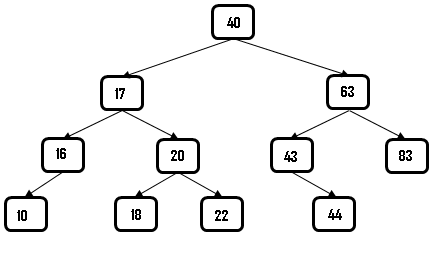
****

1. **Given the root node of a binary search tree. Write C++ code/ algorithm to display all parent nodes in ascending order where node’s value is the average of its children.**

**Assume a node of a BST is defined as follows:**

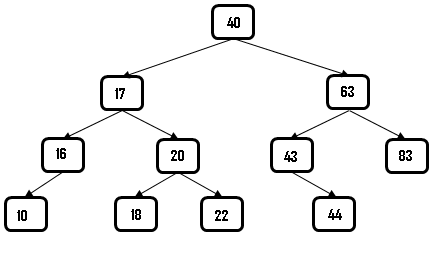
**struct nodeBST  
{  
 int data;  
 struct nodeBST \*Left;**

**struct nodeBST \*Right;  
};**

****

**Hints: If the BST as shown above is considered, your program should display 20, 40, 63 (in ascending order)**

**3.**

****

1. **Draw the BST after node value 40 is deleted.**
2. **Draw the BST after operation in (i) and adding node value 39.**

**4.**

|  |  |
| --- | --- |
| **(a)** | **Traverse the given tree using Preorder, Postorder and Backward Inorder traversals.** |
| **(b)**  **(c)** | **Construct a BST with the following data: 23 18 12 20 44 35 52. (Figure only)**  **Assume a node of a BST is defined as follows:**  **struct nodeBST {  int data;  struct nodeBST \*Left;**  **struct nodeBST \*Right; };**  **Using the above structure write down the C++ code/algorithm to find the second largest node of a BST.** |

**5.**

|  |  |
| --- | --- |
| **(a)** | **Traverse the given tree using Inorder, Preorder and Postorder traversals.**  **ques-18** |
| **(b)** | **Define a binary search tree (BST). Write down the algorithm to add a node to BST.** |

**6.**

|  |  |
| --- | --- |
| **(a)** | **Traverse the given tree using Inorder, Preorder and Postorder traversals.**  **An example binary tree** |
| **(b)** | **Write down an algorithm / function to clear (delete) all nodes of a binary tree.** |
| **(c)** | ***Definition***  ***Tree?***  ***Full binary tree/ complete binary tree?*** |

**7.**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **a)**  **b)** | **In the given binary tree, using array store the all nodes indicating their location.**   |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | | ***8*** | ***?*** | ***?*** | ***?*** | ***?*** | ***?*** | ***?*** | ***?*** | ***?*** | ***?*** | ***?*** | ***?*** | ***?*** | ***?*** | ***?*** |     **What the different possible ways a node can be detected from a BST?**  **In a BST, deleting a node that has both children can be performed in two different ways. Explain them. 4+4**  **Write a algorithm / code insert a node to a binary search tree. 8** |