

9/12/25

- WAP
- Construct binary search tree
 - Transverse the tree (Preorder, inorder and postorder)
 - Display the elements in the tree.

```
a) Struct Node *Insert (Struct Node *root, int value) {  
    if (root == NULL) {  
        return createNode(value);  
    }  
    if (value < root->data) {  
        root->left = Insert (root->left, value);  
    }  
    else if (value > root->data) {  
        root->right = Insert (root->right, value);  
    }  
    return root;  
}
```

```
b) Void Inorder (Struct Node *root) {  
    if (root == NULL)  
        return;  
    Inorder (root->left);  
    printf ("%d", root->data);  
    Inorder (root->right);  
}
```

```
Void preorder (Struct Node *root) {  
    if (root == NULL)  
        return;  
    printf ("%d", root->data);  
    preorder (root->left);  
}
```

```

preorder(root->right); }
void postorder(struct Node* root) {
    if (root == NULL)
        return;
    postorder(root->left);
    postorder(root->right);
    printf("%d ", root->data); }

```

```

c) void display(struct Node* root) {
    printf("Binary search tree (Inorder)");
    inorder(root);
    printf("\n");
    printf("Preorder");
    preorder(root);
    printf("\n");
    printf("Postorder");
    postorder(root);
    printf("\n"); }

```

O/p.

1. Insert
2. Inorder Traversal
3. Preorder Traversal
4. Postorder Traversal
5. Display
6. Exit

Enter choice : 1

How many values to insert : 9

Enter value 1 : 8

Enter value 2 : 3

Enter value 3 : 10

Enter value 4 : 1

Enter value 5 : 6

Enter value 6 : 4

Enter value 7 : 7

Enter value 8 : 14

Enter value 9 : 13

Enter choice : 2

1 3 4 6 7 8 10 13 14

Enter choice : 3

8 3 1 6 4 7 10 14 13

Enter choice : 4

1 4 7 6 3 13 14 10 8

Enter choice : 5

BINARY SEARCH TREE

Inorder Traversal : 1 3 4 6 7 8 10 13 14

Preorder Traversal : 8 3 1 6 4 7 10 14 13

Postorder Traversal : 1 4 7 6 3 13 14 10 8