# Week - 8

5-2-24

Write a program to construct a binary search tree
Traverse and display using inorder, postorder, preorder.

```
#include <stdio.h>
#include <stdlib.h>
struct Node
{
    int data;
    struct Node* left;
    struct Node* right;
};
struct Node * newNode(int data)
{
     struct Node *node = (struct Node *)malloc(sizeof(struct Node));
     node → data = data;
     node → left = node → right = NULL;
     return node;
}
struct Node * insert (struct Node *root, int data)
{
     if(root == NULL)
          return newNode(data);
     if( data <= root → data)
          root → left - insert(root → left, data);
     else
          root → right = insert (root → right, data);
}
void inorder (struct Node* temp)
{
     if (temp == NULL)
          return;
     inorder (temp → left);
     printf("%d ", temp → data);
     inorder (temp → right);
}
```

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

void main()
{
        struct Node *root = NULL;
        int data, choice;
        root = insert (root, 20);
        root = insert (root, 10);
        root = insert (root, 5);
        root = insert (root, 15);
        root = insert (root, 40);
        root = insert (root, 30);
        root = insert (root, 50);
       printf("\n the inorder traversal is: \n");
       inorder(root);
       printf("\n");
       printf("\n the preorder traversal is: \n");
       preorder (root);
       printf("\n");
```

15/2/24

```
        printf("\n the postorder traversal is : \n");
        postorder(root);
        printf("\n");
}
```

Output:-

```
 the inorder traversal is:
 5  10  15  20  30  40  50

the preorder traversal is:
 20  10  5  15  40  30  50

the postorder traversal is:
5  15  10  30  50  40  20
```