DARSHAN NINGTHOUJAM
1BM19CS041

Binary Search Tree

```
#include <stdio.h>
#include <malloc.h>

typedef struct Node
{
 int data;
 struct Node *left, *right;
 node;
}
node *newNode (int data)
{
 node *temp = (node*) malloc (sizeof(node));
 temp -> data = data;
  temp -> left = temp -> right = NULL;
  return temp;
}
void inorder (node *root)
 {
  if (root != NULL)
  { inorder (root -> left);
   printf ("%d", root -> data);
   inorder (root -> right);
  }
 }
 void preorder (node *root)
  {
   if (root != NULL)
   {
    printf("%d ", root -> data);
    inorder (root -> left);
    inorder (root -> right);
   }
  }
```

```
void postorder(node *root)
{
 if(root != NULL)
  {inorder(root->left);
   inorder(root->right);
   printf("%d ", root->data);
}
}

node *insert(node *node, int data)
{
 if(node == NULL);
  return newNode(data);
 if(data< node->data)
  node->left = insert(node->left, data);
 else
  node->right = insert(node->right, data);
 return node;
}
 int main(void)
 {node *root = NULL;
  root = insert(root, 20);
  root = insert(root, 40);
  root = insert(root, 52);
  root = insert(root, 9);
  root = insert(root, 96);
  root = insert(root, 98);
  root = insert(root, 108);
  root = insert(root, 690);
  root = insert(root, 4);
  printf("Inorder traversal: \n");
  inorder(root);
```

```
printf("Preorder Traversal: \n");
preorder(root);
printf("Postorder traversal: \n");
postorder(root);
}
```