ADITHI GIRIMAJI
1BM19CS005

Lab-10

```
#include<stdio.h>
#include<stdlib.h>
typedef struct BST
{ int data;
  struct BST *left;
  struct BST *right;
}node;
node *create();
void insert(node *, node*);
void preorder(node*)
void inorder(node*);
void postorder(node*);
int main()
{ char ch;
  node *root=NULL, *temp;
  int choice;
  while(1)
  { printf("1.construct \n2.Preorder Traversal \n3.Inord
  traversal \n4: Postorder traversal \n5.Exit \n");
  printf("\nEnter your choice \n");
  scanf("%d", &choice);
  switch(choice)
  { case 1:
      do { temp=create();
           if(root==NULL)
           root=temp;
           else
           insert(root, temp);
           printf("Do you want to continue (y/n)?\n
           getchar();
           scanf("%c",&ch);
```

```
}while(ch=='y'||ch=='Y');
break;
case 2:
printf("\n Preorder traversal: ");
preorder(root);
break;
case 3:
printf("\n inorder traversal: ");
inorder(root);
break;
case 4:
printf("\n Post order traversal: ");
postorder(root);
break;
case 5:
exit(0);
}}
return 0;
}
node *create()
{ node *temp;
  printf("Enter the data: ");
  temp=(node *)malloc(sizeof(node));
  scanf("%d", &temp->data);
  temp->left=temp->right=NULL;
  return temp;
}
void insert(node *root, node *temp)
{ if(temp->data<root->data)
  { if(root->left!=NULL)
      insert(root->left, temp)
    else
```

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