Trisha
1BM19CS214

Binary Tree

```
struct btnode
{ int value;
   struct btnode *l;
   struct btnode *r;
 } *root = NULL , *temp = NULL;

void insert ()
 {    create ();
      if (root = NULL)
   root = temp;
      else
      search(root);
 }
void create () {
     int data;
     printf (" Enter the data ");
     scanf ("%d", &data);
     temp = (struct btnode *) malloc ( sizeof (struct btnode));
     temp→ value = data;
     temp → l = temp → r = NULL;
  }
void search ( struct btnode * t)
{
  if (( temp → value > t→ value) && (t→r != NULL))
         search ( t → r);
  else if (( temp → value > t→ value) && (t→r = = NULL))
         t→ r = temp;
  else if (( temp → value < t→ value ) && (t→l != NULL))
         search (t → l);
```

```
else if ((temp→value < t→value) && (t→l == NULL))
        t→l = temp ;
    }

void inorder (struct btnode * t)
{     if (root == NULL)
        {  printf ( " No elements to display");
           return ;
        }
        if (t→ l != NULL)
           inorder (t→l);
           printf (" %d → ", t→value);
              if (t→r != NULL)
              inorder (t→r);
        }.
void preorder (struct btnode * t)
 {     if (root == NULL)
          {  printf ("No elements to display");
             return ;
          }
      printf (" %d→ ", t→value);
        if (t→l != NULL)
           preorder (t→l);
        ~~el~~ if (t→r != NULL)
              preorder (t→r);
      }

void postorder (struct btnode *t)
   {       if (root == NULL)
             {  printf (" No elements ");
                return;
             }
```

Scanned with CamScanner

```
if ( t → l != NULL)
    postorder (t → l)
if ( t → r != NULL)
    postorder (t → r);
  printf ( "%d → " , t → value );
```