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

struct btnode
{
    int value;
    struct btnode *l;
    struct btnode *r;
}*root = NULL, *temp = NULL, *t2, *t1;
void insert();
void inorder(struct btnode *t);
void create();
void search(struct btnode *t);
void preorder(struct btnode *t);
void postorder(struct btnode *t);
void main()
{
    int ch;

    printf("\nOPERATIONS ---");
    printf("\n1 - Insert an element into tree\n");
    printf("2 - Inorder Traversal\n");
printf("3 - Preorder Traversal\n");
printf("4 - Postorder Traversal\n");
printf("5 - Exit\n");
while(1)
{
    printf("\nEnter your choice : ");
    scanf("%d", &ch);
    switch (ch)
    {
    case 1:
        insert();
        break;

    case 2:
        inorder(root);
        break;
    case 3:
        preorder(root);
        break;
    case 4:
        postorder(root);
        break;
        case 5:
            exit(0);
        default :
            printf("Wrong choice, Please enter correct choice  ");
            break;
        }
    }
}

/* To insert a node in the tree */
void insert()
{
    create();
    if (root == NULL)
        root = temp;
    else
        search(root);
}

/* To create a node */
void create()
{
    int data;


    printf("Enter data of node to be inserted : ");
    scanf("%d", &data);
    temp = (struct btnode *)malloc(sizeof(struct btnode));
    temp->value = data;
    temp->l = temp->r = NULL;
}


/* Function to search the appropriate position to insert the new node
*/
void search(struct btnode *t)
{
    if ((temp->value > t->value) && (t->r != NULL))      /* value more than
root node value insert at right */
        search(t->r);
    else if ((temp->value > t->value) && (t->r == NULL))
        t->r = temp;
    else if ((temp->value < t->value) && (t->l != NULL))    /* value less
than root node value insert at left */
        search(t->l);
    else if ((temp->value < t->value) && (t->l == NULL))
        t->l = temp;
```

```
}

/* recursive function to perform inorder traversal of tree */
void inorder(struct btnode *t)
{
    if (root == NULL)
    {
        printf("No elements in a tree to display");
        return;
    }
    if (t->l != NULL)
        inorder(t->l);
    printf("%d -> ", t->value);
    if (t->r != NULL)
        inorder(t->r);
}
```

```
/* To find the preorder traversal */
void preorder(struct btnode *t)
{
  if (root == NULL)
  {
      printf("No elements in a tree to display");
      return;
  }
  printf("%d -> ", t->value);
  if (t->l != NULL)
      preorder(t->l);
  if (t->r != NULL)
      preorder(t->r);
}
```

```
/* To find the postorder traversal */
void postorder(struct btnode *t)
{
    if (root == NULL)
    {
          printf("No elements in a tree to display ");
          return;
    }
    if (t->l != NULL)
          postorder(t->l);
    if (t->r != NULL)
          postorder(t->r);
    printf("%d -> ", t->value);
}
```