Binary

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

struct node
{ int data;
  struct node * left;
  struct node * right; };
  struct node *root = null;

void insert (int data)
{ struct node *temp = (struct node*) malloc (sizeof
                                    (struct Node));
  struct node * current;
  struct node * parent;

  temp -> data = data;
  temp -> left = null;
  temp -> right = null;

  if (root == = null)
  { root = temp; }

  else
  { current = root;
    parent = null;
```

```
while (1)
{ parent = current;
  if (data < parent->data)
{ current = current -> left;

 if (current == null)
 { parent -> left = temp; return; } }

else
{ current = current -> right;
 if (current == null)
 { parent -> right = temp; return; }
} } } }

struct node * search (int data)
{ struct node * current = root;
  while (current -> data != data)
{ if (current != null)
   printf ("%d", current -> data);

 if (current -> data > data)
 { current = current -> left; }

else
{ current = current -> right; }
```

```
if (current == null)
{ return null; } } return current; }

void preorder_traversal (struct node* root)
{ if (root != null)
  { printf ("%d", root->data);
  preorder_traversal (root->left);
  preorder_traversal (root->right); } }

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

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

```
int main()
{ int i; int arr[7] = { 27, 14, 35, 10, 19, 31, 42 };

for (i = 0; i < 7; i++)
  { insert (arr[i]); }

i = 31;

struct node * temp2 = search(i);
if (temp2 != null)
 { printf ("%d element found", temp2->data); }

else
{ printf ("Not found"); }

printf ("Preorder");
preorder (root);

printf ("Inorder");
inorder (root);

printf ("postorder");
postorder (root);

return 0; }
```