BST

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

struct node {
  int data;
  struct node *left;
  struct node *right;
};

struct node* create(int x) {
  struct node *temp;
  temp = malloc(sizeof(struct node));
  temp -> data = x;
  temp -> left = NULL;
  temp -> right = NULL;

  return temp;
}

struct node* insert(struct node * root, int x) {
  if (root == NULL)
    return create(x);
```

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

```
    postorder(root -> right);
    printf(" %d ", root -> data);
  }
}

int main() {
  struct node *root;
  root = create(20);
  insert(root, 5);
  insert(root, 1);
  insert(root, 15);
  insert(root, 9);
  insert(root, 30);
  insert(root, 25);
  insert(root, 40);

  printf("Inorder:\n");
  inorder(root);
  printf("\n");

  printf("Preorder:\n");
  preorder(root);
  printf("\n");

  printf("Postorder:\n");
  postorder(root);
```

```
  printf("\n");


  return 0;
}
```

output