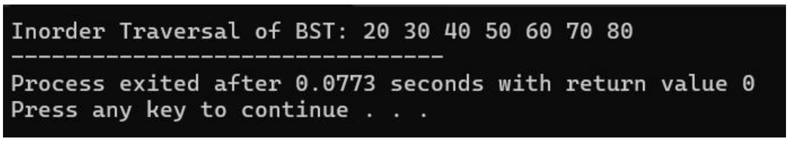
BINARY SEARCH TREE INSERTION

```
#include<stdio.h>
#include<stdlib.h>
struct node{
        int data;
        struct node* left;
        struct node* right;
};
struct node* createnode(int val){
        struct node*temp=(struct node*)malloc(sizeof(struct node));
        temp->data=val;
        temp->left=NULL;
        temp->right=NULL;
        return temp;
}
struct node* insert(struct node* root,int val){
        if(root==NULL){
                return createnode(val);
        }
        if(val<root->data)
                root->left=insert(root->left,val);
        else if(val>root->data)
                root->right=insert(root->right,val);
        return root;
}
void inorder(struct node* root){
    if (root != NULL) {
        inorder(root->left);
        printf("%d ", root->data);
        inorder(root->right);
    }
}
int main(){
    struct node* root = NULL;
    root = insert(root, 50);
    insert(root, 30);
    insert(root, 70);
    insert(root, 20);
    insert(root, 40);
    insert(root, 60);
    insert(root, 80);
    printf("Inorder Traversal of BST: ");
    inorder(root);
    return 0;
}
```

OUTPUT:



```
Inorder Traversal of BST: 20 30 40 50 60 70 80
--------------------------------
Process exited after 0.0773 seconds with return value 0
Press any key to continue . . .
```