Binary Search Tree Insertion

```
#include<stdio.h>
#include<stdlib.h>
// Basic struct of Tree
struct node
{
   int val;
   struct node *left, *right;
};
  // Function to create a new Node
struct node* newNode(int item)
{
   struct node* temp =  (struct node *)malloc(sizeof(struct node));
   temp->val = item;
   temp->left = temp->right = NULL;
   return temp;
}
  // Function print the node in inorder format, when insertion is complete
void inorder(struct node* root)
{
   if (root != NULL)
   {
      inorder(root->left);
      printf("%d \n", root->val);
      inorder(root->right);
   }
}
  // Here we are finding where to insert the new node so BST is followed
struct node* insert(struct node* node, int val)
{
//Write your code here
}
```

```
 int main()
{
   struct node* root = NULL;
   root = insert(root, 100);
   insert(root, 60);
   insert(root, 40);
   insert(root, 80);
   insert(root, 140);
   insert(root, 120);
   insert(root, 160);
        // Finally printing the tree using inorder
   inorder(root);
     return 0;
}
```