15/02/23

```
#include<stdio.h>
#include<stdlib.h>
struct node{
int data;
struct node* left;
struct node*right;
};
struct node*newNode(int data){
struct node* node= (struct node*)malloc(sizeof(struct node));
node-> data = data;
 node-> left =node ->right = NULL;
 return node;
};
struct node *insert (struct node*root, int data){
     if (root==NULL) 
           return newNode(data);
     if (data<=root->data)
          root->left = insert( root->left, data);
     else
        root -> right= insert(root->right, data);
     return root;
  }
void inorder(struct node*temp){
       if(temp == NULL)
           return;
     inorder(temp -> left);
     printf("%d\t",temp->data);
     inorder(temp-> right);

}
```

```
void preorder(struct node* temp){
    if(temp==NULL)
        return;
    printf("%d\t", temp->data);
    preorder(temp->left);
    preorder(temp->right);
}
void postorder(struct node* temp){
    if(temp==NULL)
        return;
    postorder(temp->left);
    postorder(temp->right);
    printf("%d\t", temp->data);
}
int main() {
    struct node* root = NULL;
    int data;
    root = insert(root, 1);
    root = insert(root, 2);
    root = insert(root, 3);
    root = insert(root, 4);
    printf("\n inorder traverse:");
    inorder(root);
    printf("\n preorder traverse:");
    preorder(root);
    printf("\n postorder traverse:");
    postorder(root);
    return 0;
}
```

[illegible] 13/2/24

output:-
```
inorder traverse: 1  2  3 4
preorder traverse: 1  2  3  4 ✓
postorder traverse: 4 3  2  1 ✓
```