```c
#include <stdio.h>
#include <stdlib.h>
typedef struct BSTnode
{
    int data;
    struct BSTnode *left;
    struct BSTnode *right;
} BSTnode;

#define NEWNODE (BSTnode*)malloc(sizeof(BSTnode));

BSTnode *insert(BSTnode *T, int x)
{
    BSTnode *temp;
    if (T == NULL)
    {
        temp = NEWNODE;
        temp->data = x;
        temp->left = NULL;
        temp->right = NULL;
        return temp;
    }
    else if (x < T->data)
        T->left = insert(T->left, x);
    else if (x >= T->data)
        T->right = insert(T->right, x);
    return T;
}

void preorder(BSTnode* T)
{
    if(T!=NULL)
    {
        printf("%d ",T->data);
        preorder(T->left);
        preorder(T->right);
    }
}

void inorder(BSTnode* T)
{
    if(T!=NULL)
    {
        inorder(T->left);
        printf("%d ",T->data);
        inorder(T->right);
    }
}

void postorder(BSTnode* T)
{
    if(T!=NULL)
    {
        postorder(T->left);
```

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



int main()
{
    BSTnode *root = NULL;
    int n, i, x, choice;
    printf("\nEnter how many nodes:");
    scanf("%d", &n);
    for (i = 1; i <= n; i++)
    {
        printf("\nEnter data:");
        scanf("%d", &x);
        root = insert(root, x);
    }
    do
    {
        printf("\n1:Preorder\n2:Inorder\n3:Postorder\n4:Insert a node\n5:Exit");
        printf("\nEnter your choice:");
        scanf("%d", &choice);
        switch(choice)
        {
            case 1: preorder(root);
                    break;
            case 2: inorder(root);
                    break;
            case 3: postorder(root);
                    break;
            case 4: printf("\nEnter data:");
                    scanf("%d",&x);
                    root = insert(root,x);
        }
    } while (choice != 5);

    return 0;
}
```