Start here | binarysearchtree.c

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

struct Node {
    int data;
    struct Node *left, *right;
};

struct Node *createNode(int value){
    struct Node *newNode = (struct Node*)malloc(sizeof(struct Node));
    newNode->data = value;
    newNode->left = newNode->right = NULL;
    return newNode;
}
struct Node *insert(struct Node* root, int value){
    if(root == NULL) return createNode(value);
    if(value < root->data){
        root->left = insert(root->left, value);
    }
    else if(value > root->data){
        root->right = insert(root->right, value);
    }
    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);
    }
}
```

```c

int main(){
    struct Node *root = NULL;
    int n, value;
    printf("Enter number of elements to insert into Binary Search Tree: ")
    scanf("%d",&n);

    printf("Enter %d elements:\n",n);
    for(int i =0;i<n;i++){
        scanf("%d",&value);
        root = insert(root,value);
    }

    printf("\nInorder Traversal: ");
    inorder(root);

    printf("\nPreorder Traversal: ");
    preorder(root);

    printf("\nPostorder Traversal: ");
    postorder(root);

    return 0;
}

```

C:\Users\admin\Desktop\1BM

```
Enter number of elements to insert into Binary Search Tree: 6
Enter 6 elements:
12
-23
-3
100
56
95

Inorder Traversal: -23 -3 12 56 95 100
Preorder Traversal: 12 -23 -3 100 56 95
Postorder Traversal: -3 -23 95 56 100 12
Process returned 0 (0x0)   execution time : 36.795 s
Press any key to continue.
```