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

struct BST {
   int data;
   struct BST *left, *right;
};
typedef struct BST NODE;

/* Insert */
NODE* insert(NODE *root, int data) {
   if (!root) {
      root = (NODE*)malloc(sizeof(NODE));
      root->data = data;
      root->left = root->right = NULL;
      return root;
   }
   if (data < root->data)
      root->left = insert(root->left, data);
   else if (data > root->data)
      root->right = insert(root->right, data);
   return root;
}

/* Search */
void search(NODE *root, int data) {
   if (!root) {
      printf("Element not found\n");
      return;
   }
   if (data < root->data)
      search(root->left, data);
   else if (data > root->data)
      search(root->right, data);
   else
      printf("Element found: %d\n", data);
}

/* Traversals */
void inorder(NODE *root) {
   if (root) {
      inorder(root->left);
      printf("%d ", root->data);
      inorder(root->right);
```

```
    }
}

void preorder(NODE *root) {
    if (root) {
        printf("%d ", root->data);
        preorder(root->left);
        preorder(root->right);
    }
}

void postorder(NODE *root) {
    if (root) {
        postorder(root->left);
        postorder(root->right);
        printf("%d ", root->data);
    }
}

int main() {
    NODE *root = NULL;
    int ch, n, data, i;

    while (1) {
        printf("\n1.Insert\n2.Search\n3.Inorder\n4.Preorder\n5.Postorder\n6.Exit\n");
        scanf("%d", &ch);

        switch (ch) {
        case 1:
            printf("Enter n: ");
            scanf("%d", &n);
            for (i = 0; i < n; i++) {
                scanf("%d", &data);
                root = insert(root, data);
            }
            break;

        case 2:
            printf("Enter element: ");
            scanf("%d", &data);
            search(root, data);
            break;

        case 3: inorder(root); break;
```

```
        case 4: preorder(root); break;
        case 5: postorder(root); break;
        case 6: exit(0);
        default: printf("Invalid choice\n");
        }
    }
}
```