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

struct Node {
char name[50];
struct Node *left;
struct Node *right;
};

struct Node* createNode(char name[]) {
struct Node *newNode = (struct Node*)malloc(sizeof(struct Node));
strcpy(newNode->name, name);
newNode->left = NULL;
newNode->right = NULL;
return newNode;
}


struct Node* insert(struct Node* root, char name[]) {
if (root == NULL) {
return createNode(name);
}

if (strcmp(name, root->name) < 0) {

root->left = insert(root->left, name);
}
else if (strcmp(name, root->name) > 0) {

root->right = insert(root->right, name);
}

}

/* Inorder traversal*/
void inorder(struct Node* root) {
if (root != NULL) {
inorder(root->left);
printf("%s ", root->name);
inorder(root->right);
}
}

/* Preorder traversal */
void preorder(struct Node* root) {
if (root != NULL) {
printf("%s ", root->name);
preorder(root->left);
preorder(root->right);
}
}
```

```
/* Postorder traversal*/
void postorder(struct Node* root) {
if (root != NULL) {
postorder(root->left);
postorder(root->right);
printf("%s ", root->name);
}
}

struct Node* searchNode(struct Node* root, char key[]) {
if (root == NULL) {
return NULL;
}
if (strcmp(root->name, key) == 0) {
return root;
}
if (strcmp(key, root->name) < 0) {
return searchNode(root->left, key);
} else {

return searchNode(root->right, key);
}
}

int main() {
struct Node* root = NULL;
int n, i;
char name[50];
char key[50];

printf("How many names do you want to insert? ");
scanf("%d", &n);
for (i = 0; i < n; i++) {
printf("Enter name %d: ", i + 1);
scanf("%s", name);
root = insert(root, name);
}

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

printf("\nPostorder Traversal: ");
postorder(root);
scanf("%s", key);
struct Node* found = searchNode(root, key);
if (found != NULL) {
printf("%s found in directory.\n", key);
} else {
printf("%s not found in directory.\n", key);
}
```

```
return 0;
}
```