exp_10/a.c

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

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

struct Node* createNode(int data) {
    struct Node* newNode = (struct Node*) malloc(sizeof(struct Node));
    newNode->data = data;
    newNode->left = newNode->right = NULL;
    return newNode;
}

void preorder(struct Node* root) {
    if (root == NULL) return;
    printf("%d ", root->data);
    preorder(root->left);
    preorder(root->right);
}

void inorder(struct Node* root) {
    if (root == NULL) return;
    inorder(root->left);
    printf("%d ", root->data);
    inorder(root->right);
}

void postorder(struct Node* root) {
    if (root == NULL) return;
    postorder(root->left);
    postorder(root->right);
    printf("%d ", root->data);
}

struct Node* insert(struct Node* root, int data) {
    if (root == NULL) return createNode(data);
    char dir;
    printf("Insert %d to left or right of %d? (l/r): ", data, root->data);
    scanf(" %c", &dir);
    if (dir == 'l')
        root->left = insert(root->left, data);
    else if (dir == 'r')
        root->right = insert(root->right, data);
    else
        printf("Invalid choice. Try again.\n");
```

```c
    return root;
}

int main() {
    struct Node* root = NULL;
    int n, val;
    printf("Enter number of nodes: ");
    scanf("%d", &n);
    if (n > 0) {
        printf("Enter root node value: ");
        scanf("%d", &val);
        root = insert(root, val);
        for (int i = 1; i < n; i++) {
            printf("Enter value of node %d: ", i + 1);
            scanf("%d", &val);
            root = insert(root, val);
        }
    }

    printf("Preorder traversal: ");
    preorder(root);
    printf("\nInorder traversal: ");
    inorder(root);
    printf("\nPostorder traversal: ");
    postorder(root);
    printf("\n");
    return 0;
}

```