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

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

struct node* createNode(int data){
    struct node *n; // creating a node pointer
    n = (struct node *) malloc(sizeof(struct node)); // Allocating
    n->data = data; // Setting the data
    n->left = NULL; // Setting the left and right children to NULL
    n->right = NULL; // Setting the left and right children to NULL
    return n; // Finally returning the created node
}

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);
    }
}

int main(){

    // Constructing the root node - Using Function (Recommended)
    struct node *p = createNode(4);
    struct node *p1 = createNode(1);
```

```
40:     struct node *p2 = createNode(6);
41:     struct node *p3 = createNode(5);
42:     struct node *p4 = createNode(2);
43:     // Finally The tree looks like this:
44:     //       4
45:     //      / \
46:     //     1   6
47:     //    / \
48:     //   5   2
49:
50:     // Linking the root node with left and right children
51:     p->left = p1;
52:     p->right = p2;
53:     p1->left = p3;
54:     p1->right = p4;
55:
56:     preOrder(p);
57:     printf("\n");
58:     postOrder(p);
59:     return 0;
60: }
61:
```