

main.c



Run

Output

Clear

```
1 #192210211
2 #include <stdio.h>
3 #include <stdlib.h>
4 struct node {
5     int item;
6     struct node* left;
7     struct node* right;
8 };
9
10 void inorderTraversal(struct node* root) {
11     if (root == NULL) return;
12     inorderTraversal(root->left);
13     printf("%d ->", root->item);
14     inorderTraversal(root->right);
15 }
16
17 void preorderTraversal(struct node* root) {
18     if (root == NULL) return;
19     printf("%d ->", root->item);
20     preorderTraversal(root->left);
21     preorderTraversal(root->right);
22 }
23 void postorderTraversal(struct node* root) {
24     if (root == NULL) return;
25     postorderTraversal(root->left);
26     postorderTraversal(root->right);
27     printf("%d ->", root->item);
28 }
29 struct node* createNode(value) {
30     struct node* newNode = malloc(sizeof(struct node));
```

```
/tmp/tYTSR8B0GQ.o
Inorder traversal
5 ->12 ->6 ->1 ->9 ->
Preorder traversal
1 ->12 ->5 ->6 ->9 ->
Postorder traversal
5 ->6 ->12 ->9 ->1 ->
```

main.c



Run

Output

Clear

```
28 }
29 struct node* createNode(value) {
30     struct node* newNode = malloc(sizeof(struct node));
31     newNode->item = value;
32     newNode->left = NULL;
33     newNode->right = NULL;
34
35     return newNode;
36 }
37 struct node* insertLeft(struct node* root, int value) {
38     root->left = createNode(value);
39     return root->left;
40 }
41 struct node* insertRight(struct node* root, int value) {
42     root->right = createNode(value);
43     return root->right;
44 }
45 int main() {
46     struct node* root = createNode(1);
47     insertLeft(root, 12);
48     insertRight(root, 9);
49     insertLeft(root->left, 5);
50     insertRight(root->left, 6);
51     printf("Inorder traversal \n");
52     inorderTraversal(root);
53     printf("\nPreorder traversal \n");
54     preorderTraversal(root);
55     printf("\nPostorder traversal \n");
56     postorderTraversal(root);
57 }
```

```
/tmp/tYTSR8B0GQ.o
Inorder traversal
5 ->12 ->6 ->1 ->9 ->
Preorder traversal
1 ->12 ->5 ->6 ->9 ->
Postorder traversal
5 ->6 ->12 ->9 ->1 ->
```