

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

```
struct node {
    int info;
```

```
    struct node * link, * rlink;
}; typedef struct node * node;
```

```
node getnode (int item) {
```

```
    Node temp = (Node) malloc (sizeof (struct node));
```

```
    temp -> info = item;
```

```
    temp -> link = temp -> rlink = NULL;
```

```
    return temp; }
```

```
Node insert (Node node, int info)
```

```
{ if (node == NULL)
```

```
    return getnode (info);
```

```
    if (info < node -> info)
```

```
        node -> link = insert (node -> link, info);
```

```
    else
```

```
        node -> rlink = insert (node -> rlink, info);
```

```
    return node; }
```

```
void preorder (Node root) {
```

```
    if (root == NULL) {
```

```
        return; }
```

```
    printf ("%d", root -> info);
```

```
    preorder (root -> link);
```

```
    preorder (root -> rlink); }
```

```
void inorder (Node root) {
```

```
if (root == NULL)
```

```
{  
    return;  
}
```

```
inorder (root -> llink);
```

```
printf ("%d -> ", root -> info);
```

```
inorder (root -> rlink); }
```

```
void postorder (Node root) {
```

```
    if (root == NULL) { return; }
```

```
    postorder (root -> llink);
```

```
    postorder (root -> rlink);
```

```
    printf ("%d -> ", root -> info); }
```

```
void display (Node root, int
```

```
    { int i;
```

```
    if (root != NULL)
```

```
    { display (root -> rlink, i+1);
```

```
      for ( i=0 ; i < 1 ; i++)
```

```
        printf ("%d\n",
```

```
        printf ("%d\n", root -> info);
```

```
        display (root -> llink, i+1);
```

```
    } }
```

```
int main ()
```

```
{ Node root = NULL;
```

```

int choice, item;
for(;;) {
    printf("\n 1. Insert\n 2. Preorder\n 3. Inorder\n 4. Postorder\n 5. Display\n 6. exit\n");
    printf("Enter choice: ");
    scanf("%d", &choice);
    switch(choice) {
        case 1: printf("Enter item to be inserted: ");
                scanf("%d", &item);
                root = insert(root, item);
                break;
        case 2: printf("Preorder traversal: ");
                preorder(root);
                break;
        case 3: printf("Inorder traversal: ");
                inorder(root);
                break;
        case 4: printf("Postorder traversal: ");
                postorder(root);
                break;
        case 5: display(root);
                break;
        case 6: exit(0);
                break;
        default: printf("Enter proper instructions !!\n");
                break;
    }
}
return 0;
}

```