```c
#include<stdio.h>
#include<stdlib.h>
struct Node
{
    struct Node *llink;
    int data;
    struct Node *rlink;
};
typedef struct Node* NODE;

NODE create()
{
    NODE newnode;
    int x;
    newnode=(NODE)malloc(sizeof(struct Node));
    printf("Enter data(-1 for no data): ");
    scanf("%d",&x);
    if(x==-1)
    return 0;
    newnode->data=x;
    printf("Enter left child of %d: \n",x);
    newnode->llink=create();
    printf("Enter right child of %d \n",x);
    newnode->rlink=create();
    return newnode;
}

void inorder(NODE head)
{
    if(head!=0)
    {
        inorder(head->llink);
        printf("%d\t\n",head->data);
        inorder(head->rlink);
    }
}

void preorder(NODE head)
{
    if(head!=0)
    {
        printf("%d\t\n",head->data);
        preorder(head->llink);
        preorder(head->rlink);
    }
}

void postorder(NODE head)
{
    if(head!=0)
    {
        postorder(head->llink);
```

```
        postorder(head->rlink);
        printf("%d\t\n",head->data);
    }
}
void display(NODE head,int i)
{
int j;
if(head!=NULL)
{
display(head->rlink,i+1);
for (j=1;j<=i;j++)
printf("   ");
printf("%d\n",head->data);
display(head->llink,i+1);
}
}

int main()
{
    NODE head=0;
    int ch;
    for(;;)
    {
    printf("1:Insert\n2:Inorder\n3:Display\n4:Preorder\n5:Postorder\n");
    printf("Enter your choice");
    scanf("%d",&ch);
    switch(ch)
    {
        case 1:head=create();
            break;
        case 2:
            inorder(head);
            break;
        case 3:
             display(head,1);
             break;
        case 4:
            preorder(head);
            break;
        case 5:
            postorder(head);
            break;
    }
    }
}
```

OUTPUT:

```
Enter your choice1
Enter data(-1 for no data): 5
Enter left child of 5:
Enter data(-1 for no data): 10
Enter left child of 10:
Enter data(-1 for no data): 11
Enter left child of 11:
Enter data(-1 for no data): -1
Enter right child of 11
Enter data(-1 for no data): -1
Enter right child of 10
Enter data(-1 for no data): 12
Enter left child of 12:
Enter data(-1 for no data): -1
Enter right child of 12
Enter data(-1 for no data): -1
Enter right child of 5
Enter data(-1 for no data): 15
Enter left child of 15:
Enter data(-1 for no data): 20
Enter left child of 20:
Enter data(-1 for no data): -1
Enter right child of 20
Enter data(-1 for no data): -1
Enter right child of 15
Enter data(-1 for no data): 25
Enter left child of 25:
Enter data(-1 for no data): -1
Enter right child of 25
Enter data(-1 for no data): -1
1:Insert
2:Inorder
3:Display
```

```
4:Preorder
5:Postorder
Enter your choice3
      25
    15
      20
  5
      12
    10
      11
1:Insert
2:Inorder
3:Display
4:Preorder
5:Postorder
Enter your choice2
11
10
12
5
20
15
25
1:Insert
2:Inorder
3:Display
4:Preorder
5:Postorder
Enter your choice5
11
12
10
20
25
15
```

```
Enter your choice5
11
12
10
20
25
15
5
1:Insert
2:Inorder
3:Display
4:Preorder
5:Postorder
Enter your choice4
5
10
11
12
15
20
25
```