

Binary Tree

```
#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", x);
    newnode->llink = create();
    printf("Enter right child of %d", x);
    newnode->rlink = create();
    return newnode;
}
```

void inorder (NODE head)

```
{
    if (head != 0)
    {
        inorder (head → link);
        printf ("%d", head → data);
        inorder (head → link);
        printf ("\n");
    }
}
```

void preorder (NODE head)

```
{
    if (head != 0)
    {
        printf ("%d", head → data);
        preorder (head → link);
        preorder (head → link);
    }
}
```

```
void postorder (NODE head)
```

```
{  
    if (head != 0)  
    {  
        postorder (head → llink);  
        postorder (head → rlink);  
        printf("%d ", 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;
```

printf("%d\n", lch);

printf("1: Enter Insert | 2: inorder | 3: display | 4: preorder | 5: postorder | 6: exit\n");

printf("Enter choice: ");

scanf("%d", &lch);

switch(lch)

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;