

```

1  #include<stdio.h>
2  #include<stdlib.h>
3  struct node
4  {
5      int data;
6      struct node* left;
7      struct node* right;
8  }*root1;
9
10 struct node *create()
11 {
12     struct node *temp;
13
14     printf("\n Enter data:");
15
16     temp=(struct node*)malloc(sizeof(struct node));
17
18     scanf("%d",&temp->data);
19
20     temp->left=temp->right=NULL;
21
22     return temp;
23
24 }
25
26 void insert(struct node *root,struct node *temp)
27 {
28     if(temp->data<root->data)
29     {
30         if(root->left!=NULL)

```

```

38
39 insert(root->left,temp);
40
41 else
42
43 root->left=temp;
44
45 }
46
47 if(temp->data>root->data)
48
49 {
50
51 if(root->right!=NULL)
52
53 insert(root->right,temp);
54
55 else
56
57 root->right=temp;
58 }
59 }
60
61
62 void Postorder(struct node* node)
63 {
64     if (node == NULL)
65         return;
66
67
68     Postorder(node->left);
69
70     Postorder(node->right);
71
72
73     printf("%d ", node->data);
74 }

```



```

76
77 void Inorder(struct node* node)
78 {
79     if (node == NULL)
80         return;
81
82
83     Inorder(node->left);
84
85     printf("%d ", node->data);
86
87
88     Inorder(node->right);
89 }
90
91
92 void Preorder(struct node* node)
93 {
94     if (node == NULL)
95         return;
96
97
98     printf("%d ", node->data);
99
100
101     Preorder(node->left);
102
103
104     Preorder(node->right);
105 }
106 int main()
107 {
108     int ch;
109     struct node *temp;
110     do
111     {

```

```

struct node *temp;
do
{
printf("1.create\n2.insert\n3.preorder\n4.postorder\n5.inorder\n6.Exit\n");
scanf("%d",&ch);
switch(ch)
{
case 1:
root1=create();
break;
case 2:
printf("enter the elem to be entered\n");
temp=(struct node*)malloc(sizeof(struct node));
scanf("%d",&temp->data);
insert(root1,temp);
break;
case 3:
Preorder(root1);
printf("\n");
break;
case 4:
Postorder(root1);
printf("\n");
break;
case 5:
Inorder(root1);
printf("\n");
break;
case 6:
break;
default:
printf("wrong entry");
}
}while(ch!=6);

```

```
1.create
2.insert
3.preorder
4.postorder
5.inorder
6.Exit
1
    Enter data:4
1.create
2.insert
3.preorder
4.postorder
5.inorder
6.Exit
2
enter the elem to be entered
5
1.create
2.insert
3.preorder
4.postorder
5.inorder
6.Exit
2
enter the elem to be entered
6
1.create
2.insert
3.preorder
4.postorder
5.inorder
6.Exit
2
```

enter the elem to be entered

7

1.create

2.insert

3.preorder

4.postorder

5.inorder

6.Exit

2

enter the elem to be entered

8

1.create

2.insert

3.preorder

4.postorder

5.inorder

6.Exit

2

enter the elem to be entered

9

1.create

2.insert

3.preorder

4.postorder

5.inorder

6.Exit

3

4 5 6 7 8 9

1.create

2.insert

3.preorder

4.postorder

5.inorder

6.Exit

```
2
enter the elem to be entered
9
1.create
2.insert
3.preorder
4.postorder
5.inorder
6.Exit
3
4 5 6 7 8 9
1.create
2.insert
3.preorder
4.postorder
5.inorder
6.Exit
4
9 8 7 6 5 4
1.create
2.insert
3.preorder
4.postorder
5.inorder
6.Exit
5
4 5 6 7 8 9
1.create
2.insert
3.preorder
4.postorder
5.inorder
6.Exit
```