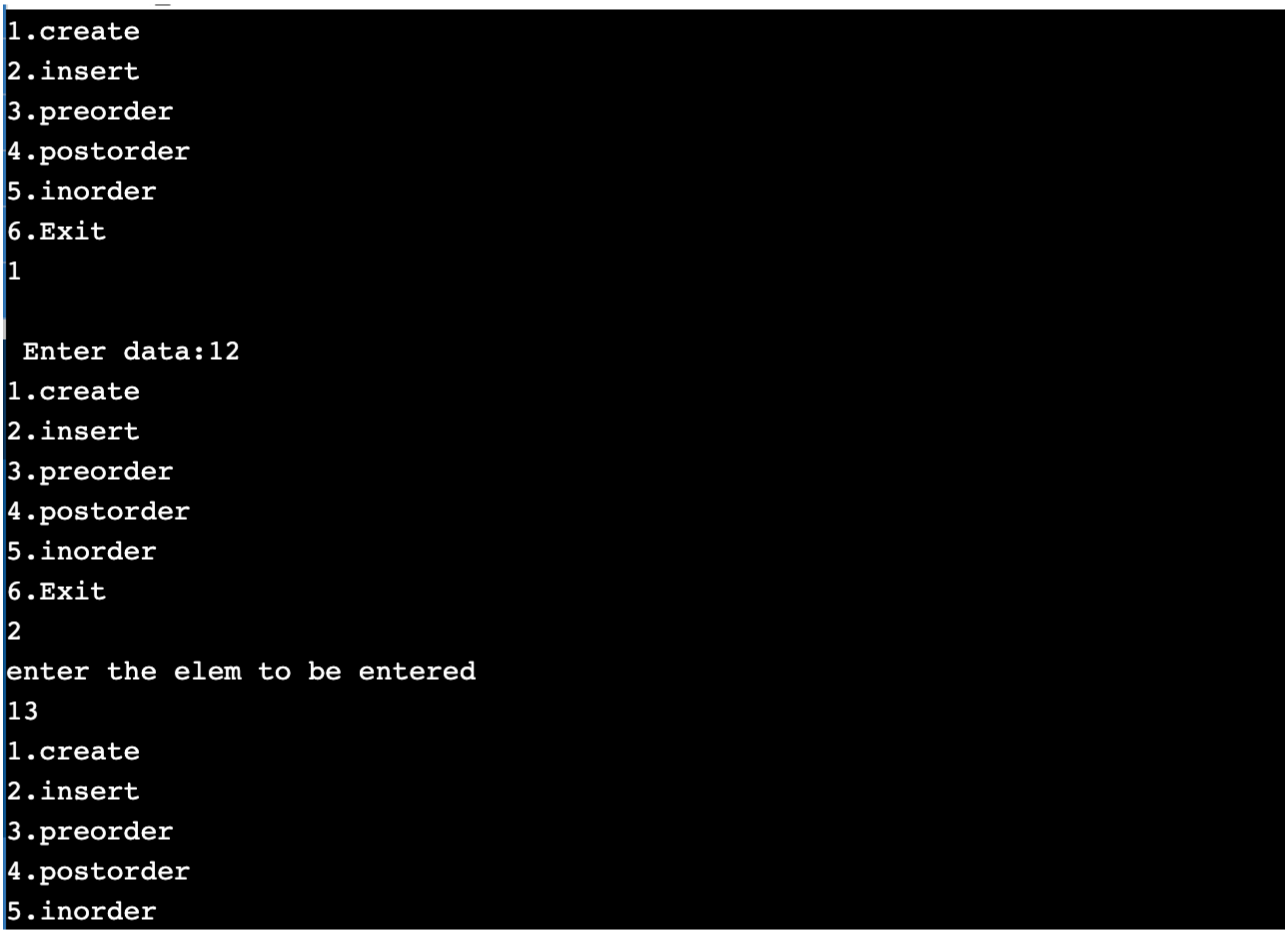
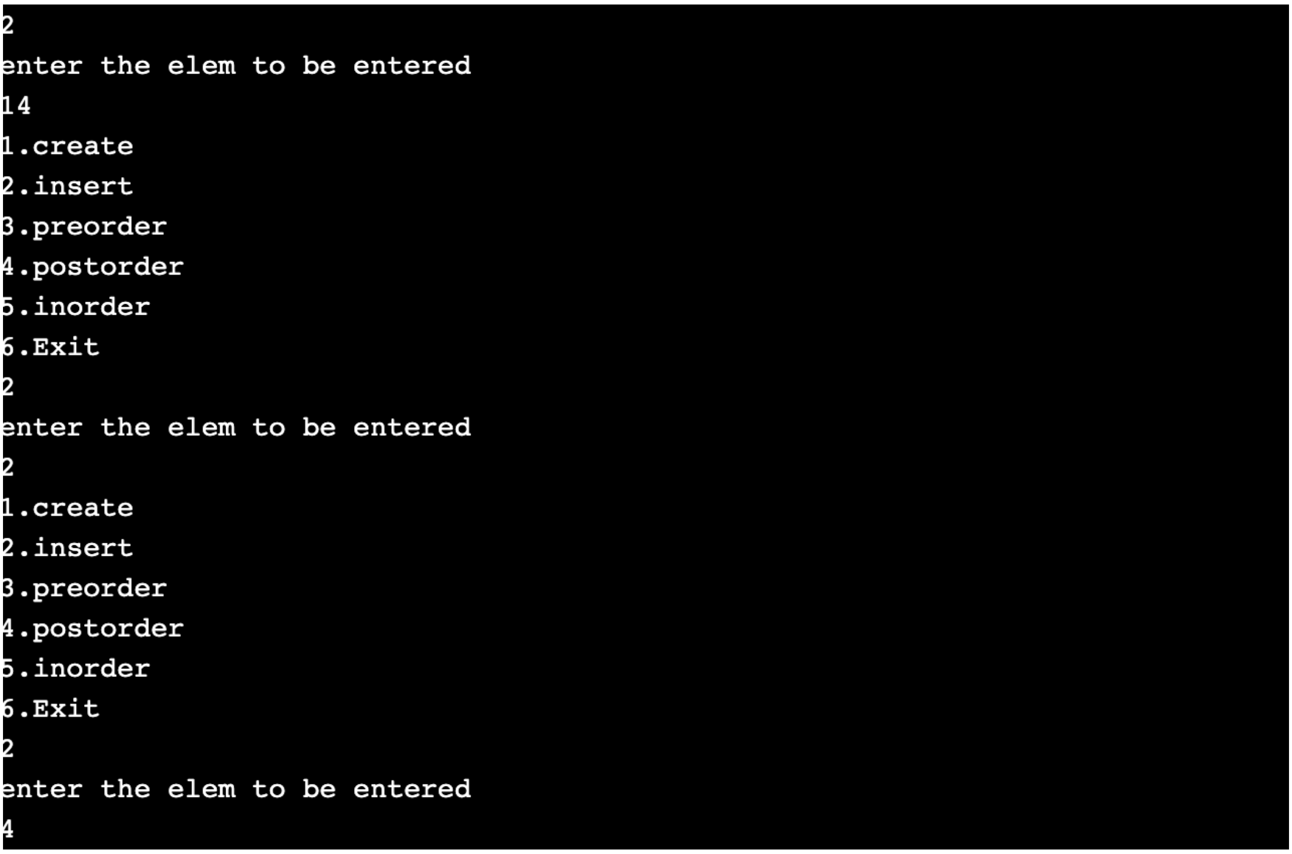
```c
#include<stdio.h>
#include<stdlib.h>
struct node
{
int data;
struct node* left;
struct node* right;
}*root1;

struct node *create()

{

struct node *temp;

printf("\n Enter data:");

temp=(struct node*)malloc(sizeof(struct node));

scanf("%d",&temp->data);

temp->left=temp->right=NULL;

return temp;

}

void insert(struct node *root,struct node *temp)

{


if(temp->data<root->data)
```

```

{

if(root->left!=NULL)

insert(root->left,temp);

else

root->left=temp;

}

if(temp->data>root->data)

{

if(root->right!=NULL)

insert(root->right,temp);

else

root->right=temp;
}
}


void Postorder(struct node* node)
{
if (node == NULL)
return;


Postorder(node->left);

Postorder(node->right);


printf("%d ", node->data);
}


void Inorder(struct node* node)
{
if (node == NULL)
return;


Inorder(node->left);

printf("%d ", node->data);


Inorder(node->right);
}


void Preorder(struct node* node)
{
if (node == NULL)
return;


printf("%d ", node->data);


Preorder(node->left);


Preorder(node->right);
}
```

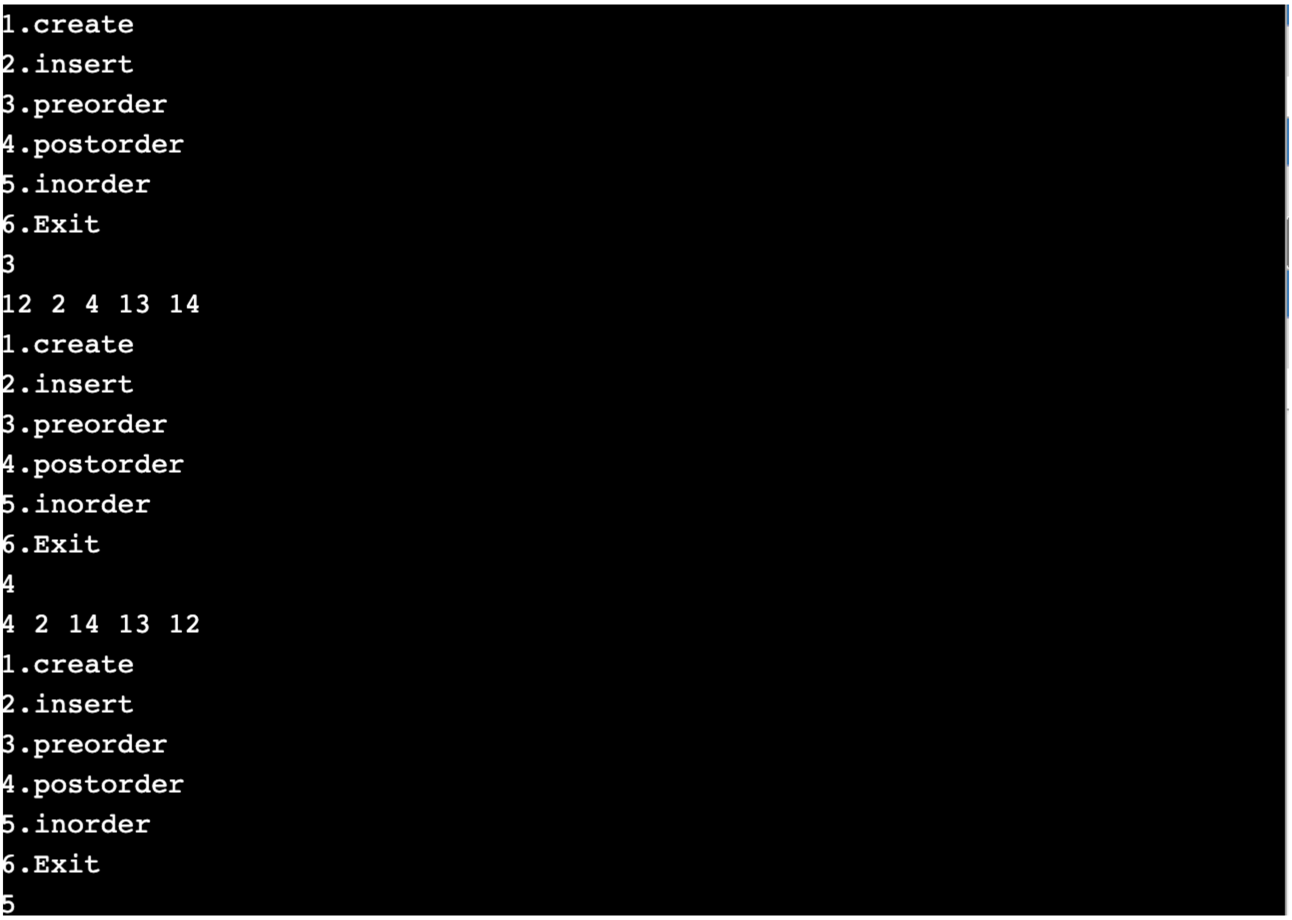
```
int main()
{
int ch;
struct node *temp;
do
{
printf("1.create\n2.insert\n3.preorder\n4.postorder\n5.inorder\n6.Exit\n");
scanf("%d",&ch);
switch(ch)
{
case 1:
root1=create();
break;
case 2:
printf("enter the elem to be entered\n");
temp=(struct node*)malloc(sizeof(struct node));
scanf("%d",&temp->data);
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);


}
```



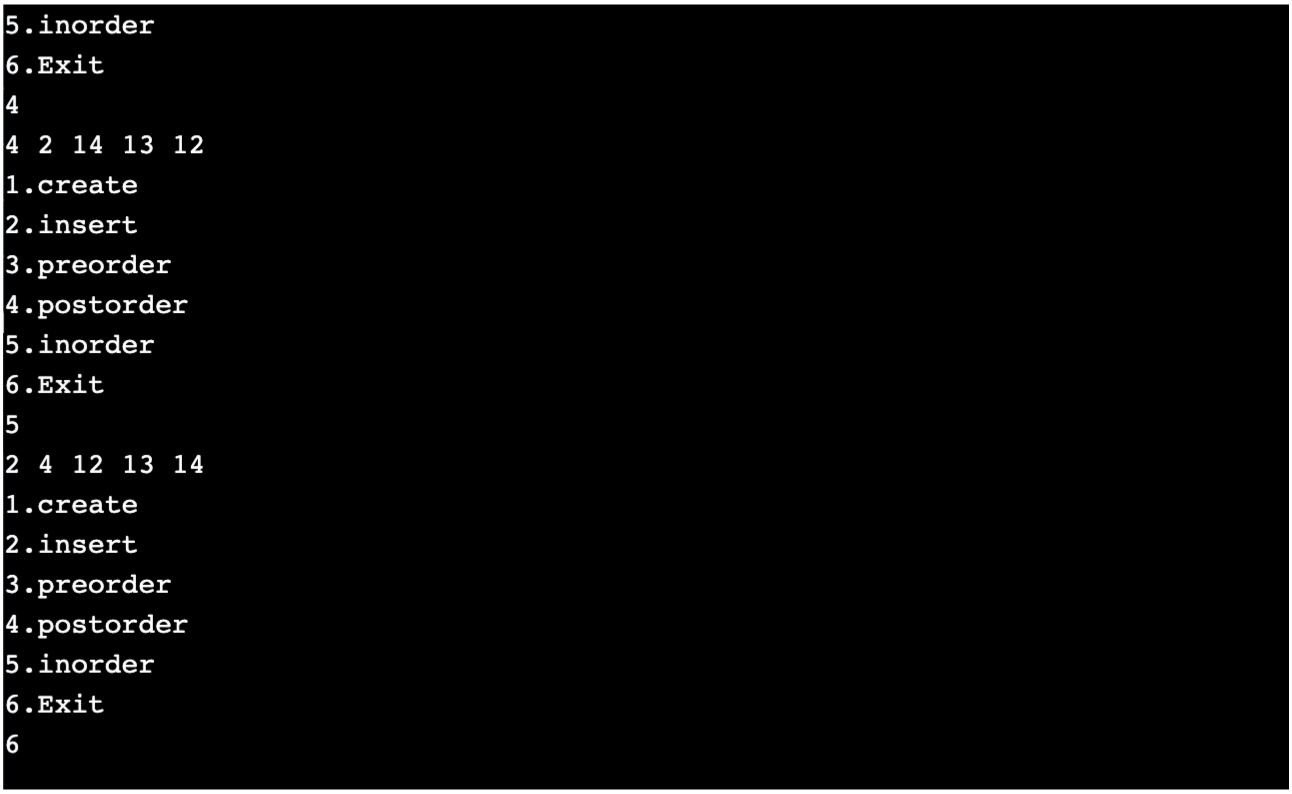
```
1.create
2.insert
3.preorder
4.postorder
5.inorder
6.Exit
1

 Enter data:12
1.create
2.insert
3.preorder
4.postorder
5.inorder
6.Exit
2
enter the elem to be entered
13
1.create
2.insert
3.preorder
4.postorder
5.inorder
```



```
2
enter the elem to be entered
14
1.create
2.insert
3.preorder
4.postorder
5.inorder
6.Exit
2
enter the elem to be entered
2
1.create
2.insert
3.preorder
4.postorder
5.inorder
6.Exit
2
enter the elem to be entered
4
```



```
1.create
2.insert
3.preorder
4.postorder
5.inorder
6.Exit
3
12 2 4 13 14
1.create
2.insert
3.preorder
4.postorder
5.inorder
6.Exit
4
4 2 14 13 12
1.create
2.insert
3.preorder
4.postorder
5.inorder
6.Exit
5
```



```
5.inorder
6.Exit
4
4 2 14 13 12
1.create
2.insert
3.preorder
4.postorder
5.inorder
6.Exit
5
2 4 12 13 14
1.create
2.insert
3.preorder
4.postorder
5.inorder
6.Exit
6
```