INPUT:

```
#include <iostream>
using namespace std;
class BTT{
	struct node{
		int data;
		node *lc;
		node *rc;
		int lflag;
		int rflag;
	}*root,*header;
	int count=0;
public:
	BTT()
	{
		root=NULL;
		header = new node;
		header->lc=header->rc=header;
		header->lflag=header->rflag=0;   //flag value 0 indicate no child but thread exists
	}
	void create();
	void preorder_BTT();
	void postorder_BTT();
	void inorder_BTT();
	void menu();
};


void BTT::create()
{
	char ans='y';
	do{
		count++;
		node *temp=new node;
		cout<<"\nEnter element: ";
		cin>>temp->data;
		temp->lflag=temp->rflag=0;
		if(root==NULL)   //Only executes the first time,
		{
			root=temp;
			header->lc=root;
			header->lflag=header->rflag=1;
			root->lc=root->rc=header;
		}
		else
		{
			node *curr;
			curr=root;
			while(1)
			{
				if(temp->data < curr->data)
```

```
                                        {
                                                if(curr->lflag==0)
                                                {
                                                        temp->rc=curr;
                                                        temp->lc=curr->lc;
                                                        curr->lc=temp;
                                                        curr->lflag=1;
                                                        break;
                                                }
                                                else
                                                        curr=curr->lc;
                                        }
                                        else if(temp->data > curr->data)
                                        {
                                                if(curr->rflag==0)
                                                {
                                                        temp->lc=curr;
                                                        temp->rc=curr->rc;
                                                        curr->rc=temp;
                                                        curr->rflag=1;
                                                        break;
                                                }
                                                else
                                                        curr=curr->rc;
                                        }
                                }
                        }
                        cout<<"\nDo You want to continue?(y/n): ";
                        cin>>ans;
        }while(ans=='y' || ans=='Y');
}

void BTT::preorder_BTT()
{
        if(header->lc==header)
                cout<<"\nBinary threaded tree is empty!";
        else
        {
                node *t;
                t=root;
                cout<<t->data<<" ";
                while(1)
                {
                        while(t->lflag==1)
                        {
                                t=t->lc;
                                cout<<t->data<<" ";
                        }
                        while(t->rflag!=1)
                                t=t->rc;
                        if(t==header)
```

```
                                        break;
                                t=t->rc;
                                cout<<t->data<<" ";
                        }
                }
}

void BTT::postorder_BTT()
{
        if(header->lc==header)
                cout<<"\nBinary Threaded tree is empty.";
        else
        {
                int arr[count],i=-1;
                node *t=root;
                arr[++i]=t->data;
                while(1)
                {
                        while(t->rflag==1)
                        {
                                t=t->rc;
                                arr[++i]=t->data;
                        }
                        if(t->lflag==0 && t->rflag==0)
                                while(t->lflag!=1)
                                        t=t->lc;
                        if(t->lflag==1)
                        {
                                if(t==header)
                                        break;
                                t=t->lc;
                                arr[++i]=t->data;
                        }
                }
                for(i=count-1;i>=0;i--)
                        cout<<arr[i]<<" ";
        }
}

void BTT::inorder_BTT()
{
        if(header->lc==header)
                cout<<"\nBinary Threaded tree is empty.";
        else
        {
                int flag=0;
                node *t=root;
                while(1)
                {
                        while(t->lflag==1)
                                t=t->lc;
```

```
                        cout<<t->data<<" ";
                        if(t->rflag==1)
                                t=t->rc;
                        else
                        {
                                while(t->rflag!=1)
                                {
                                        t=t->rc;
                                        if(t==header)
                                        {
                                                flag=1;
                                                break;
                                        }
                                        cout<<t->data<<" ";
                                }
                                if(flag==1)
                                        break;
                                t=t->rc;
                        }
                }
        }
}

void BTT::menu()
{
        int ch,flag=1;
        do{
        cout<<"\n\t\tMenu for BTT";
        cout<<"\n1.Create/Insert";
        cout<<"\n2.Pre-order traversal";
        cout<<"\n3.Post-order traversal";
        cout<<"\n4.In-order traversal";
        cout<<"\n5.Exit";
        cout<<"\nEnter your choice: ";
        cin>>ch;
        switch(ch)
        {
                case 1 :create();
                                break;
                case 2: preorder_BTT();
                                break;
                case 3: postorder_BTT();
                                break;
                case 4: inorder_BTT();
                                break;
                case 5: flag=0;
                                break;
                default: cout<<"\nInvalid Input.";
                                 break;
        }
        if(flag==0)
```

```
                    break;
            }while(1);
}

int main() {
            BTT obj;
            obj.menu();
            return 0;
}
```

OUTPUT:

```
Menu for BTT
1.Create/Insert
2.Pre-order traversal
3.Post-order traversal
4.In-order traversal
5.Exit
Enter your choice: 1
Enter element: 30
Do You want to continue?(y/n): y
Enter element: 21
Do You want to continue?(y/n): y
Enter element: 56
Do You want to continue?(y/n): y
Enter element: 11
Do You want to continue?(y/n): y
Enter element: 65
Do You want to continue?(y/n): n
Menu for BTT
1.Create/Insert
2.Pre-order traversal
3.Post-order traversal
4.In-order traversal
5.Exit
Enter your choice: 2
30 21 11 56 65
                    Menu for BTT
1.Create/Insert
2.Pre-order traversal
3.Post-order traversal
4.In-order traversal
5.Exit
Enter your choice: 3
11 21 65 56 30
                    Menu for BTT
1.Create/Insert
2.Pre-order traversal
3.Post-order traversal
4.In-order traversal
5.Exit
Enter your choice: 4
```

11 21 30 56 65

Menu for BTT

1.Create/Insert
2.Pre-order traversal
3.Post-order traversal
4.In-order traversal
5.Exit
Enter your choice: 4