```
//Name: 6. Threaded Binary Tree
#include <iostream>
 using namespace std;
 struct ThreadedNode {
   int data;
   ThreadedNode *le ;
   ThreadedNode *right;
   bool le Thread;
   bool rightThread;
   ThreadedNode(int val) : data(val), le (nullptr), right(nullptr),
le Thread(true), rightThread(true) {}
 };
 class ThreadedBinaryTree {
 private:
   ThreadedNode *root;
   void insert(ThreadedNode *&node, int val) {
      if (!node) {
         node = new ThreadedNode(val);
      } else if (val < node->data) {
         insert(node->le , val);
      } else {
         insert(node->right, val);
      }
   }
```

```
void thread(ThreadedNode *&node) {
   if (!node) return;
   if (node->le ) {
      thread(node->le );
   }
   if (node->right) {
      thread(node->right);
   }
   if (node->le && !node->le->rightThread) {
      node->le->right = node;
      node->le->rightThread = true;
   }
   if (node->right && !node->right->le Thread) {
      node->right->le = node;
      node->right->le Thread = true;
   }
}
void inOrderHelper(ThreadedNode *node) {
   if (!node) return;
   while (node) {
      while (node && !node->le Thread) {
         node = node->le ;
      }
      if (!node) return; // Check for null a er le traversal
```

```
            cout << node->data << " ";
            while (node && node->rightThread) {
                node = node->right;
                cout << node->data << " ";
            }
            if (node) node = node->right;
        }
    }
    void preOrderHelper(ThreadedNode *node) {
        if (!node) return;
        cout << node->data << " ";
        if (!node->le Thread) preOrderHelper(node->le );
        if (!node->rightThread) preOrderHelper(node->right);
    }
public:
    ThreadedBinaryTree() : root(nullptr) {}
    void insert(int val) {
        insert(root, val);
    }
    void thread() {
        thread(root);
    }
    void inOrder() {
        cout << "In-order Traversal: ";
```

```
        inOrderHelper(root);
        cout << endl;
    }
    void preOrder() {
        cout << "Pre-order Traversal: ";
        preOrderHelper(root);
        cout << endl;
    }
};
int main() {
    ThreadedBinaryTree tbt;
    int baseElements[] = {5, 3, 7, 2, 4, 6, 8};
    for (int val : baseElements) {
        tbt.insert(val);
    }
    tbt.thread();
    tbt.inOrder();
    tbt.preOrder();
    return 0;
}
```