/\* C++ program to convert a Binary Tree to

Threaded Tree \*/

#include <bits/stdc++.h>

using namespace std;

struct Node

{

int key;

Node \*left, \*right;

bool isThreaded;

};

Node \*createThreaded(Node \*root)

{

if (root == NULL)

return NULL;

if (root->left == NULL &&

root->right == NULL)

return root;

if (root->left != NULL)

{

Node\* l = createThreaded(root->left);

l->right = root;

l->isThreaded = true;

}

if (root->right == NULL)

return root;

return createThreaded(root->right);

}

Node \*leftMost(Node \*root)

{

while (root != NULL && root->left != NULL)

root = root->left;

return root;

}

void inOrder(Node \*root)

{

if (root == NULL) return;

Node \*cur = leftMost(root);

while (cur != NULL)

{

cout << cur->key << " ";

if (cur->isThreaded)

cur = cur->right;

else

cur = leftMost(cur->right);

}

}

Node \*newNode(int key)

{

Node \*temp = new Node;

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

temp->key = key;

return temp;

}

int main()

{

Node \*root = newNode(1);

root->left = newNode(2);

root->right = newNode(3);

root->left->left = newNode(4);

root->left->right = newNode(5);

root->right->left = newNode(6);

root->right->right = newNode(7);

createThreaded(root);

cout << "Inorder traversal of created "

"threaded tree is\n";

inOrder(root);

return 0;

}