#include <iostream>

using namespace std;

struct Node

{

int data;

Node \*left, \*right;

};

struct Node\* getNode(int data)

{

struct Node \*newNode = new Node;

newNode->data = data;

newNode->left = newNode->right = NULL;

return newNode;

}

void preorderTraversal(Node\*);

void inorderTraversal(Node \*root, vector<int>& arr)

{

if (root == NULL)

return;

inorderTraversal(root->left, arr);

arr.push\_back(root->data);

inorderTraversal(root->right, arr);

}

void BSTToMinHeap(Node \*root, vector<int> arr, int \*i)

{

if (root == NULL)

return;

root->data = arr[++\*i];

BSTToMinHeap(root->left, arr, i);

BSTToMinHeap(root->right, arr, i);

}

void convertToMinHeapUtil(Node \*root)

{

vector<int> arr;

int i = -1;

inorderTraversal(root, arr);

BSTToMinHeap(root, arr, &i);

}

void preorderTraversal(Node \*root)

{

if (!root)

return;

cout << root->data << " ";

preorderTraversal(root->left);

preorderTraversal(root->right);

}

int main()

{

struct Node \*root = getNode(4);

root->left = getNode(2);

root->right = getNode(6);

root->left->left = getNode(1);

root->left->right = getNode(3);

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

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

convertToMinHeapUtil(root);

cout << "Preorder Traversal:" << endl;

preorderTraversal(root);

return 0;

}