#include <iostream>

using namespace std;

struct node

{

int value;

node \*left, \*right;

};

struct node\* getnode(int value)

{

struct node \*newnode = new node;

newnode->value = value;

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->value);

inorderTraversal(root->right, arr);

}

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

{

if (root == NULL)

return;

root->value = 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->value << " ";

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;

}