#include<bits/stdc++.h>

using namespace std;

struct HangHoa {

int ma;

string ten, ngay;

float gia;

};

typedef struct Node {

HangHoa data;

Node \*left, \*right;

};

typedef struct Node\* Tree;

Tree root;

void nhapHH(HangHoa &s) {

cout << "Nhap ma hang hoa: "; cin >> s.ma;

cout << "Nhap ten hang hoa: "; cin.ignore(); getline(cin, s.ten);

cout << "Nhap ngay xuat: "; cin >> s.ngay;

cout << "Nhap gia: "; cin >> s.gia;

cout << endl;

}

void xuatHH(HangHoa s) {

cout << s.ma << " " << s.ten << " " << s.ngay << " " << s.gia << endl;

}

void KhoiTao(Tree &root) {

root = NULL;

}

Node\* TaoNut(HangHoa s) {

Node \*p = new Node();

if(p == NULL) {

cout << "Khong du bo nho" << endl;

return NULL;

}

p->data = s;

p->left = p->right = NULL;

return p;

}

void chen(Tree &root, Node \*p) {

if(root == NULL) root = p;

else {

if(root->data.ma > p->data.ma) chen(root->left, p);

else if(root->data.ma < p->data.ma) chen(root->right, p);

}

}

void duyet(Tree root) {

if(root != NULL) {

duyet(root->left);

xuatHH(root->data);

duyet(root->right);

}

}

Node\* TimKiem(Tree root, int x) {

if(root != NULL) {

if(root->data.ma == x) return root;

else if(root->data.ma > x) return TimKiem(root->left, x);

else return TimKiem(root->right, x);

}

return NULL;

}

int main() {

int n;

do {

cout << "Nhap so luong: "; cin >> n;

} while(n <= 0);

Tree root;

KhoiTao(root);

for(int i = 0; i < n; i++) {

HangHoa s; nhapHH(s);

Node \*p = TaoNut(s);

if(p != NULL)

chen(root, p);

}

cout << "Sau khi nhap" << endl;

duyet(root);

int x;

cout << "Nhap ma can tim: "; cin >> x;

Node \*p = TimKiem(root, x);

if(p != NULL) {

cout << "Hang hoa can tim" << endl;

xuatHH(p->data);

}

else

cout << "Khong co hang hoa can tim";

}