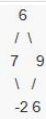
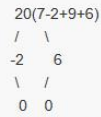


题目描述：

给出一个二叉树如下图所示：



请由该二叉树生成一个新的二叉树，它满足其树中的每个节点将包含原始树中的左子树和右子树的和。



左子树表示该节点左侧叶子节点为根节点的一颗新树；右子树表示该节点右侧叶子节点为根节点的一颗新树

输入描述: 2行整数，第1行表示二叉树的中序遍历，第2行表示二叉树的前序遍历，以空格分割

例如：

7 -2 6 6 9

6 7 -2 9 6

输出描述: 1行整数，表示求和树的中序遍历，以空格分割

例如：

输出1

-2 0 20 0 6

补充说明：

示例 1

输入：

-3 12 6 8 9 -10 -7

8 12 -3 6 -10 9 -7

输出：

0 3 0 7 0 2 0

说明：

```
#include <iostream>
```

```
#include <string>
```

```
#include <vector>
```

```
using namespace std;
```

```
void recursion(vector<int>& midorder, int l, int r, vector<int>& preorder,
               int l2, int r2) {
```

```
    if (l > r)
```

```
        return;
```

```
    int root = preorder[l2];
```

```
    int index = -1;
```

```
    for (int i = l; i <= r; ++i) {
```

```
        if (midorder[i] == root) {
```

```
            index = i;
```

```
            break;
```

```
        }
```

```
    }
```

```
    int sum = 0;
```

```
    for (int i = l; i <= r; ++i) {
```

```
        if (i != index) {
```

```
            sum += midorder[i];
```

```

    }
}
midorder[index] = sum;
int left_len = index - l;
recursion(midorder, l, index - 1, preorder, l2 + 1, l2 + left_len);
recursion(midorder, index + 1, r, preorder, l2 + left_len + 1, r2);
}

int main() {
    string str;
    getline(cin, str);
    vector<int> vec1, vec2;
    string tmp;
    for (int i = 0; i < str.length(); ++i) {
        if (!isspace(str[i])) {
            tmp += str[i];
        } else {
            vec1.push_back(atoi(tmp.c_str()));
            tmp.clear();
        }
    }
    vec1.push_back(atoi(tmp.c_str()));
    getline(cin, str);
    tmp.clear();
    for (int i = 0; i < str.length(); ++i) {
        if (!isspace(str[i])) {
            tmp += str[i];
        } else {
            vec2.push_back(atoi(tmp.c_str()));
            tmp.clear();
        }
    }
    vec2.push_back(atoi(tmp.c_str()));
    recursion(vec1, 0, vec1.size() - 1, vec2, 0, vec2.size() - 1);
    for (int i = 0; i < vec1.size(); ++i)
        cout << vec1[i] << " ";
    return 0;
}

```