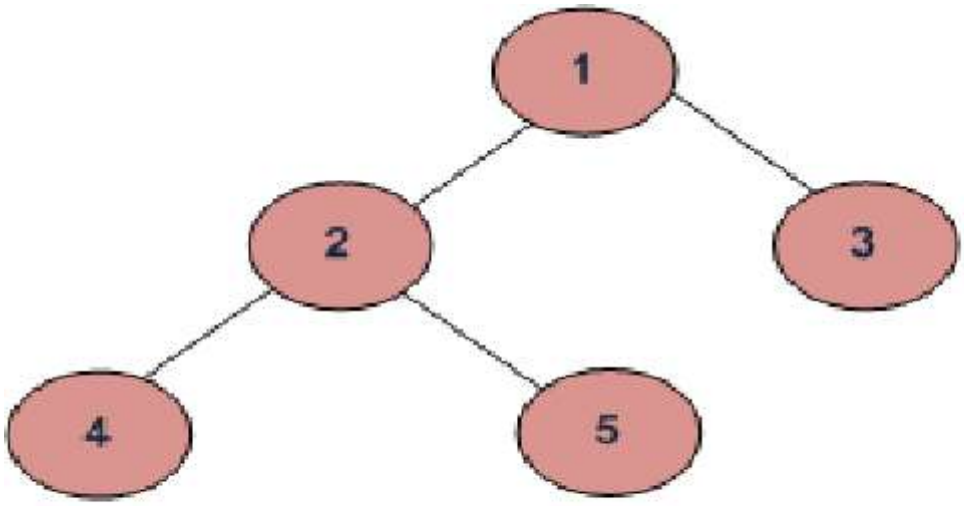
Description

Unlike linear data structures (Array, Linked List, Queues, Stacks, etc) which have only one logical way to traverse them, binary tree can be traversed in different ways.
In depth first traversal applications, a binary tree could be traversed in three different ways: Preorder, In order and Postorder. For example, in the In order traversal, we first traverse the left subtree, then the root, and finally the right subtree, hence we could obtain the order of nodes in this traversal.

Here is an example of three different traversals:
(a) Preorder (Root, Left, Right) : 1 2 4 5 3
(b) Inorder (Left, Root, Right) : 4 2 5 1 3
(c) Postorder (Left, Right, Root) : 4 5 2 3 1



Given the result of inorder and preorder traversal, can you get the result of postorder traversal?

Input

The first line of input is an integer T (1 <= T <= 200) indicating the number of test cases.
Each test case will follow the format shown below:
The first line: One integer N (1 <= N<= 100) indicating the number of tree nodes (numbered from 1 to N) in the binary tree.
The second line: N integers n1, n2, ..., nN showing the result of preorder traversal.
The third line: N integers m1, m2, ..., mN showing the result of inorder traversal.

The second line: N integers h1, h2, ..., hN showing the result of preorder traversal.
The third line: N integers m1, m2, ..., mN showing the result of inorder traversal.

Output

For each test case, print a single line containing N integers showing the result of postorder traversal for each test case, separate two integers by one space.

Sample

Sample input

```
2
8
1 2 4 7 3 5 6 8
4 7 2 1 5 3 8 6
5
1 2 4 5 3
4 2 5 1 3
```

Sample output

```
7 4 2 5 8 6 3 1
4 5 2 3 1
```

Constraint and hint

In the second test case, by definition the first element in the preorder (1 2 4 5 3) must be the root of the binary tree, i.e. node {1} is the root, hence combining with the inorder (4 2 5 1 3), we conclude that nodes {4 2 5} are from the left subtree, and {3} is from the right subtree. And we reconstruct the left subtree and right subtree using recursion. Once we finish reconstructing the binary tree, it is not difficult to print the postorder of the binary tree.

C++ 17
GCC 10.2.0

C++ (NOI)

```
1  #include <iostream>
2  using namespace std;
3
4  struct Node {
```