

```

#include <stdio.h>
#include <stdlib.h>
struct TreeNode
{
    int id;
    struct TreeNode * left;
    struct TreeNode * right;
};
int main()
{
    while(1)
    {
        int number_of_nodes=0;
        int i;
        int pre[1000];
        int in[1000];

        int weight[1000];
        struct TreeNode* currentNode;
        struct TreeNode * root = NULL;
        struct TreeNode* tmp;
        scanf("%d",&number_of_nodes);

        if(number_of_nodes==-1)
        {break;}

        for(i = 0; i< number_of_nodes; i++)
        { scanf("%d",&pre[i]);}
        for(i = 0; i< number_of_nodes; i++)
        {
            scanf("%d",&in[i]);
            weight[in[i]]=i;
        }
        for(i =0;i<number_of_nodes;i++)
        {
            int nodeId = pre[i];
            currentNode = (struct TreeNode*)malloc(sizeof(struct TreeNode));
            currentNode->id = nodeId;
            currentNode->left=NULL;
            currentNode->right = NULL;

            if(root==NULL)
            { root = currentNode; }
        }
    }
}

```

```

else
{
    tmp = root;
    while(tmp!=NULL)
    {
        if(weight[nodeId]<weight[tmp->id])
        {
            if(tmp->left==NULL)
            { tmp->left= currentNode;tmp=NULL; }
            else
            { tmp=tmp->left; }
        }
        else
        {
            if(tmp->right==NULL)
            { tmp->right = currentNode;tmp=NULL;}
            else
            { tmp=tmp->right;}
        }
    }
} //end while
} //end else
currentNode=NULL;
} //end for
printPost (root, root);
printf("\n");
}
return 0;
}

```

```

int printPost(struct TreeNode* currentNode)
{
    if(currentNode!=NULL&&root!=NULL)
    {
        if(currentNode->left!=NULL)
        {printPost (root, currentNode->left);}
        if(currentNode->right!=NULL)
        {printPost (root, currentNode->right);}
        printf("%d ",currentNode->id);
    }
    return 0;
}

```