4. Merge two binary trees.

```
class Solution {
public:
    TreeNode* mergeTrees(TreeNode* root1,
                         TreeNode* root2) {

    if (root1 == NULL && root2 == NULL)
        return NULL;

    if (root1 == NULL)
        return root2;

    if(root2 == NULL)
        return root1;

    TreeNode* merged = new TreeNode(root1->val
                                    + root2->val);

    merged->left = mergeTrees(root1->left,
                              root2->left);

    merged->right = mergeTrees(root1->right,
                               root2->right);

    return merged;
    }
};
```

Output:

case 1:

Input -

root1 =

[1, 3, 2, 5]

root2 =

[2, 1, 3, null, 4, null, 7]

Output -

[3, 4, 5, 5, 4, null, 7]

Expected

[3, 4, 5, 5, 4, null, 7]

Scanned with CamScanner

case 2 :

Input :

root1 =

[1]

root2 =

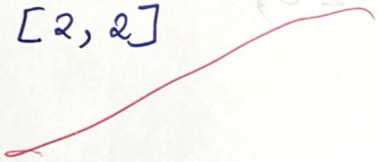
[1, 2]

Output :

[2, 2]

Expected :

[2, 2]



Scanned with CamScanner