Lab - 2.

Leetcode problem - 2583.

Kth Largest Sum in a binary tree.

```
int height(struct TreeNode* root) {
    if(root == NULL) {
        return 0;
    }
    else {
        int lheight = height(root->left);
        int rheight = height(root->right);
        if(lheight > rheight) {
            return lheight + 1;
        }else {
            return rheight + 1;
        }
    }
}

void dfs(struct TreeNode* root, int level, long long* sums) {
    if(root == NULL) {
        return;
    }
    sums[level] = sums[level] + root->val;
    if(root->left) {
        dfs(root->left, level+1, sums);
    }
    if(root->right) {
        dfs(root->right, level+1, sums);
    }
}
```

```
long long KthLargestLevelSum(struct TreeNode* root, int k) {
    int h = height(root);
    if(k>h) {
        return -1;
    }
    long long* sums = (long long*)calloc(h, sizeof(long long));
    dfs(root, 0, sums);

    for(int i=0; i<h-1; i++) {
        for(int j=0; j<h-i; j++) {
            if(sums[j] < sums[j+1]) {
                long long temp = sums[j];
                sums[j] = sums[j+1];
                sums[j+1] = temp;
            }
        }
    }
    long long largest = 0;
    largest = sums[k-1];
    free(sums);
    return largest;
}
```

9/8/24

O/P:

root = [5,8,9,2,1,3,7,4,6]

k = 2

output = 13.

root = [1,2,null,3]

k = 1

output = 3.