Practical – 19

AIM: Binary Tree Level Order Traversal Given the root of a binary tree, return the level order traversal of its nodes' values. (i.e., from left to right, level by level).

* Program

class class Solution {

vector<vector<int>> a;

vector<int> sub;

int level;

public:

int findLevel(TreeNode\* root){

if(root == NULL)

return 0;

else{

int lheight = findLevel(root->left);

int rheight = findLevel(root->right);

if(lheight > rheight)

return (lheight+1);

else

return (rheight+1);

}

}

vector<vector<int>> levelOrder(TreeNode\* root) {

level = findLevel(root);

for(int i = 1; i <= level; i++){

storeVal(root, i);

a.push\_back(sub);

sub.clear();

}

return a;

}

void storeVal(TreeNode\* root, int i){

if(root==NULL)

return;

else if(i == 1){

sub.push\_back(root->val);

}

else{

storeVal(root->left, i-1);

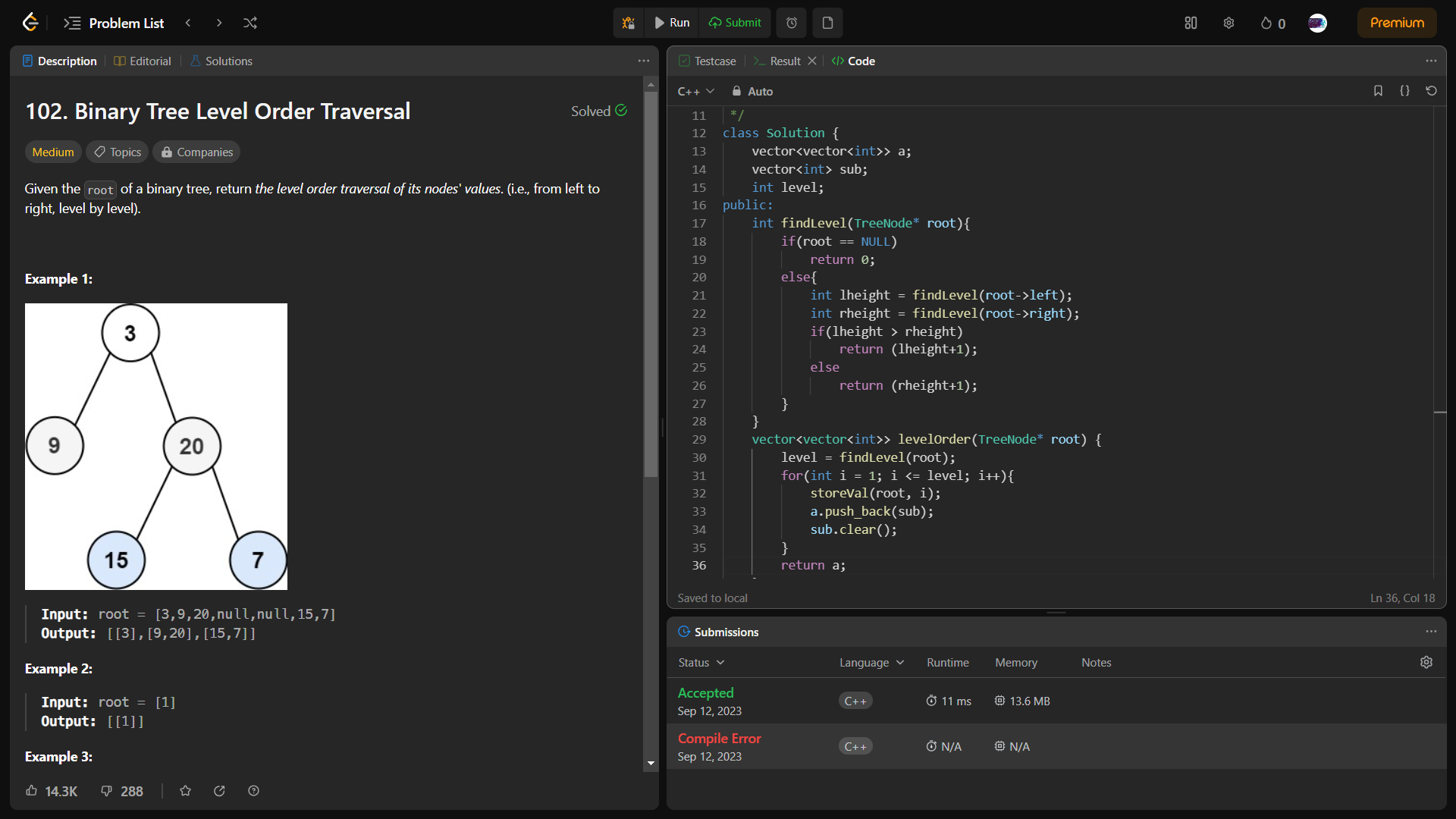
storeVal(root->right, i-1);

}

}

};

* Output



\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_

Student Signature Faculty Signature Marks