# **Experiment-2.1**

**Student Name: Aayush Gurung** 

Branch: CSE Semester: 6

**Subject Name: CC LAB** 

**UID:20BCS5323** 

Section/Group:DM\_607(A)

Date of Performance: 10-03-2023

**Subject Code: 20CSP-351** 

### Aim:

To demonstrate the concept of Trees

## **Objective**

#### **Problem 1**:

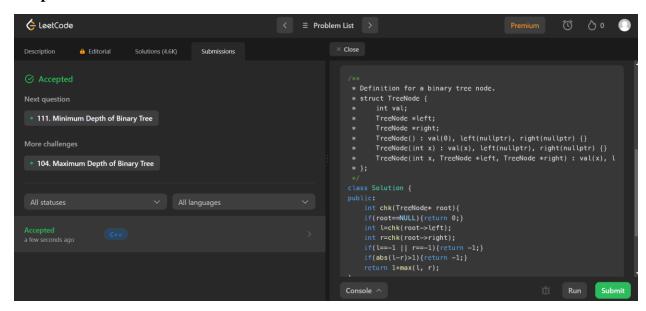
Balanced binary tree

#### Code:

```
/**
 * Definition for a binary tree node.
 * struct TreeNode {
 * int val;
 * TreeNode *left;
 * TreeNode *right;
 * TreeNode() : val(0), left(nullptr), right(nullptr) {}
 * TreeNode(int x) : val(x), left(nullptr), right(nullptr) {}
 * TreeNode(int x, TreeNode *left, TreeNode *right) : val(x), left(left), right(right) {}
 * };
 */
 class Solution {
 public:
    int chk(TreeNode* root) {
      if(root==NULL) {return 0;}
      int l=chk(root->left);
      int r=chk(root->right);
      if(l==-1 || r==-1) {return -1;}
      return 1+max(1, r);
 }
 bool isBalanced(TreeNode* root) {
    return chk(root)!=-1;
  }
```

```
};
};
```

## **Output:**



#### **Problem 2:**

Path sum

#### Code:

```
* Definition for a binary tree node.
 * struct TreeNode {
       TreeNode *left;
       TreeNode *right;
       TreeNode(int x, TreeNode *left, TreeNode *right) : val(x), left(left),
class Solution {
public:
    bool solve(TreeNode* root,int targetSum ,int &curr){
        if(root==NULL) return false;
                curr+=root->val;
         if(curr==targetSum && root->left==NULL && root->right==NULL) return true;
        bool left =solve(root->left,targetSum,curr);
        bool right=solve(root->right, targetSum, curr);
        curr-=root->val;
        return left || right;
   bool hasPathSum(TreeNode* root, int targetSum) {
        if(root==NULL) return false;
        int cr=0;
      bool ans=solve(root,targetSum,cr);
      return ans;
```

## **Output:**

bool right=solve(root->right, targetSum, curr);

Submit