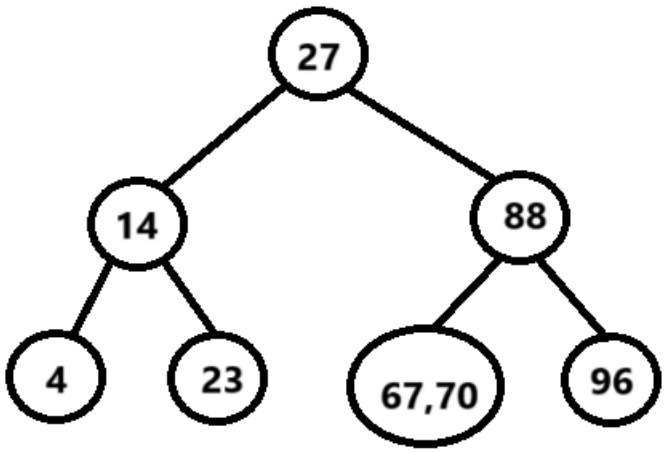
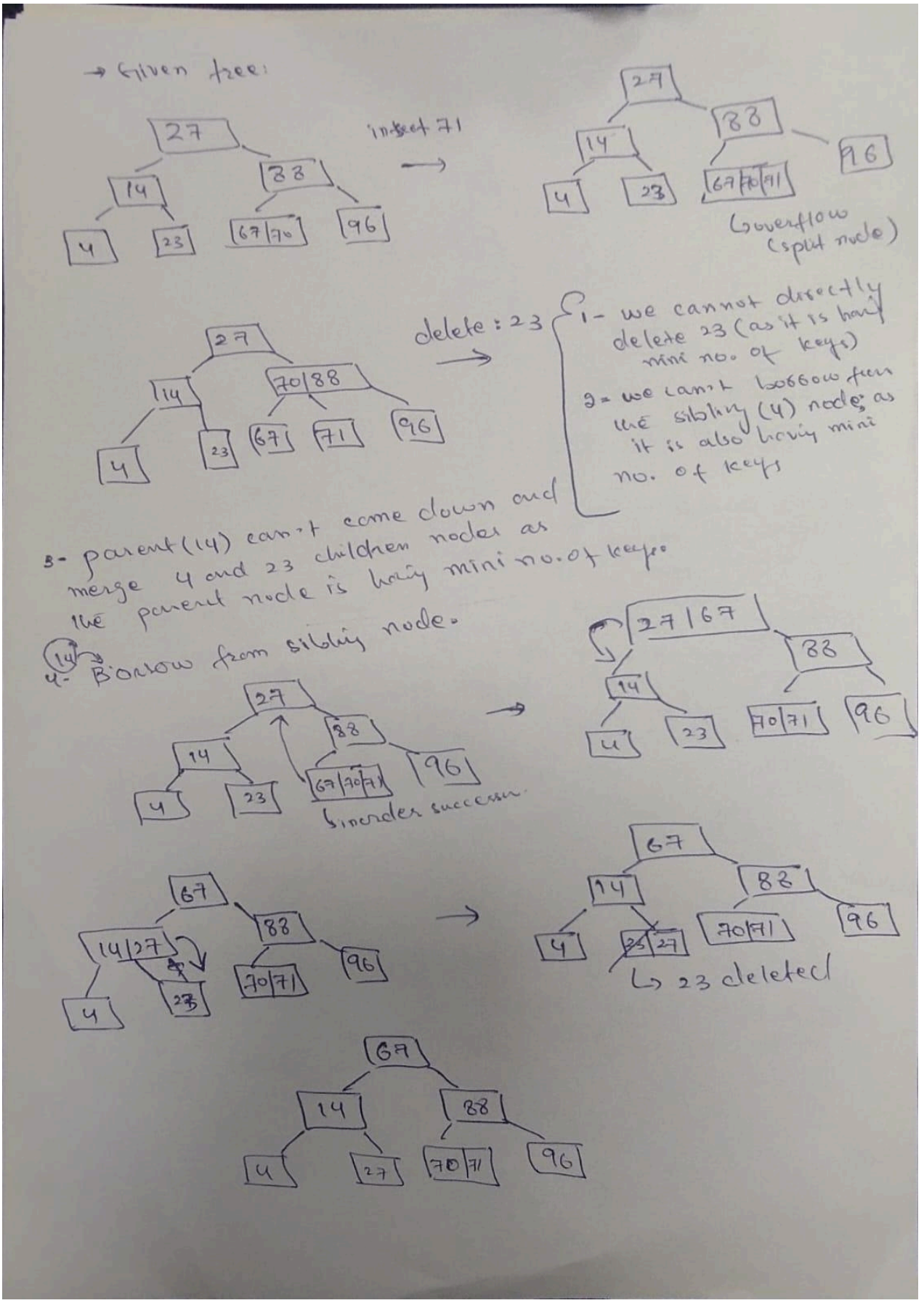
Name and ID: _____ Section: _____
Marks: 15

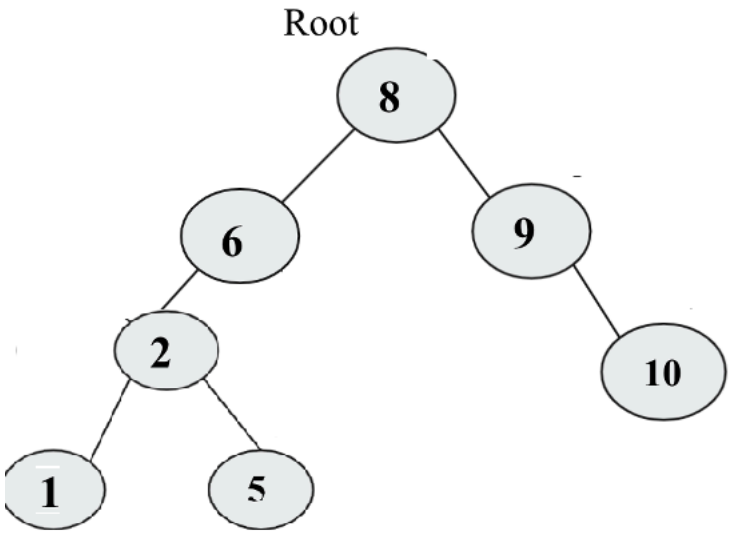
Question#01 [4+5 Marks]

a. Given the 2-3 tree below, insert 71 and delete 23. Show each step of the process clearly and the final resulting tree by drawing each step clearly.

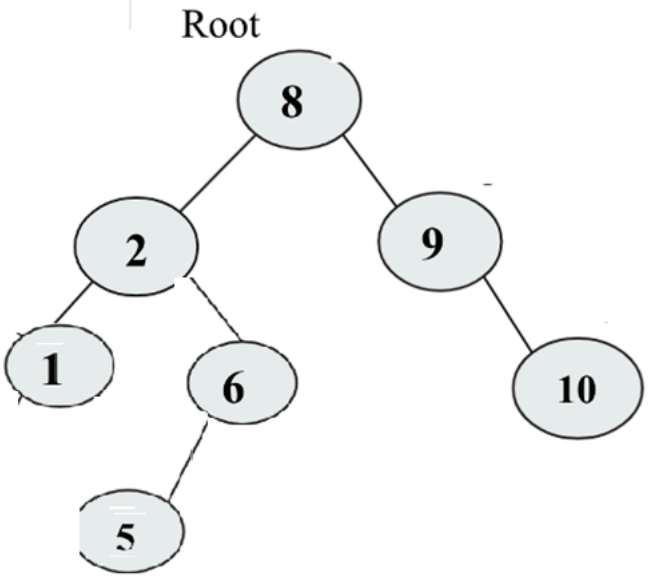




a. Find which node is imbalance in the following tree and which AVL rotation is used to balance the node. Show rotation dry run and write C++ function for that rotation case.



Rotation case: LL imbalance (right rotation)



Right rotation

```
Node* LL_rotation(Node* node) {
        Node* child = node->left_node;
        node->left_node = child->right_node;
        child->right_node = node;

        node->height = max(get_height(node->left_node), get_height(node->right_node)) + 1;
        child->height = max(get_height(child->left_node), get_height(child->right_node)) + 1;

        return child;
    }
```

Question#02 [6 Marks]

a. Write a recursive function that takes a Binary Search Trees (BST) root pointer and return size of the given tree. [4+1]
   i. int SizeOfTheTree(const BTNode *tree)
   Size of a Binary Tree is defined as the numbers of node in the tree. We can count the node recursively using the following code.

```
int SizeOfTheTree(BTNode* root) {
        int size=0;
        // Base case: empty tree has 0 node in the tree
        if (root == 0)
                return size;
        else {
                // recur for count the node in the left and right subtree
                // add them together to get the total nodes ->size
                size +=SizeOfTheTree(root->left)
                size +=SizeOfTheTree(root->right)
                return size;
        }
}
```

   ii. What will be the time complexity of this operation? O(n)

b. What are the prefix forms of the expression? [1]
   A+B*(C-D)/(P-R)
   Prefix form: +A/*B-CD-PR