4+5= 9/15

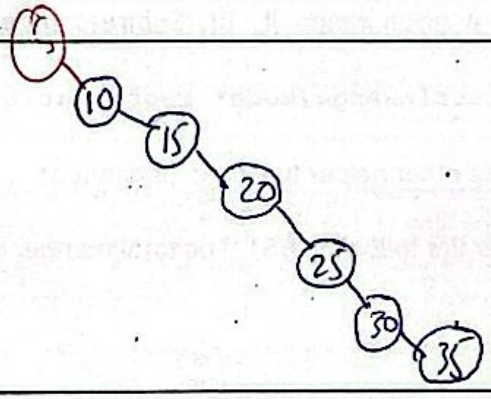
National University of Computer and Emerging Sciences (Lahore)			
Course:	Applied Programming	Code:	CS-0319
Section:	MSCS-2A	Semester:	Spring 2024
Time:	25 minutes	TotalMarks:	~~10~~ 15
Date:		Roll no:	23L-8025
Name:	Ameer Hamza		

Question#1: (4/5) **[5]**

Consider the following list of values to be inserted into the Binary Search Tree (BST): 5, 10, 15, 20, 25, 30, 35.

Draw the Binary Search Tree that results from inserting the above list of values in order.

Tree after insertion:



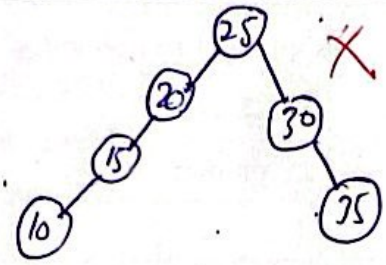
3/3.5

Consider the following piece of C++ code:

```
TreeNode* leftRotate(TreeNode* root) {
    TreeNode* newRoot = root->right;
    root->right = newRoot->left;
    newRoot->left = root;
    return newRoot;
}
TreeNode* RotationThrice(TreeNode* root) {
    for (int i = 0; i < 3; ++i) {root = leftRotate(root);}
    return root;
}
```

Suppose the function `RotationThrice(TreeNode* root)` is called for the root node of the above drawn tree. Redraw the updated tree below:

Updated Tree:



Question#2: (5/10) [2+3]

You are provided with an unordered array of integer values. Your task is to create an integer Binary Search Tree (BST) from this array using an insert function:

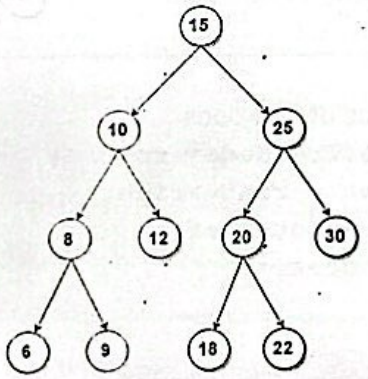
1) `Node* insertIntoBST(int * arr, int size)`

Once the BST is constructed, count the number of subtrees within it recursively where all nodes lie within a given range [L, R] (Subtrees here refer to any node and its descendants):

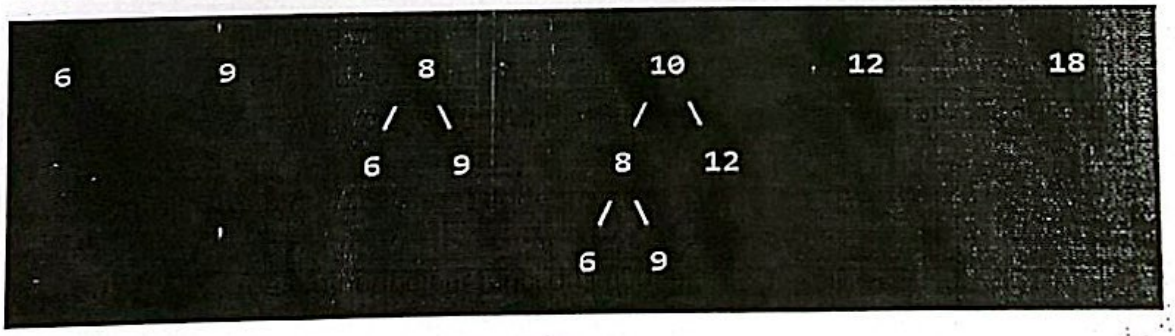
2) `int countSubtreesInRange(Node* root, int L, int R)`

NOTE: you may create other helper functions (if needed)

For Example: Consider the following BST. The total number of subtrees with nodes in range [5, 20] is 6.



The subtrees are:



No.

Rough Work

(2)

int CountSubtree in Rang (node * root,
int L, int R)

we have to create a function.

```
int count-Range(node * root, int low,
                int high)
{
    if (root == Null)
        return 0;
    ,
    if (root < low)
    {
        root = root -> right
    return CountSubtree(node * root, int L, R);
    }
    else if (root > high)
    {
        root = root -> left ;
    return countSubtree(node * root, int, R);
    }
    return root ;
```

①

4.5/5

undefined var.

should not be changing the value of root as reference is lost

But concept is right.

Part No. | Rough Work

Ameer Hamza

23L-8025

9/15

Question #02

(1)

```
node *insertintoBst (int *arr, int size)
{
  X int *array[0] = root ;        undefined in scope.

  for (int i=0 ; i<size ; i++)  ①
  {
    if (array[i] < root)
    {
  X   root->left = array[i];
    }                              have to initialize node.
                                   Type mismatch.
    else                           int != node
    {
  X   root->right = array[i];
    };
```

1/5

##
we Done using for loop
with Compare of array[i] indexes
with root ->data.