4 + 0.5 = 4.5/15

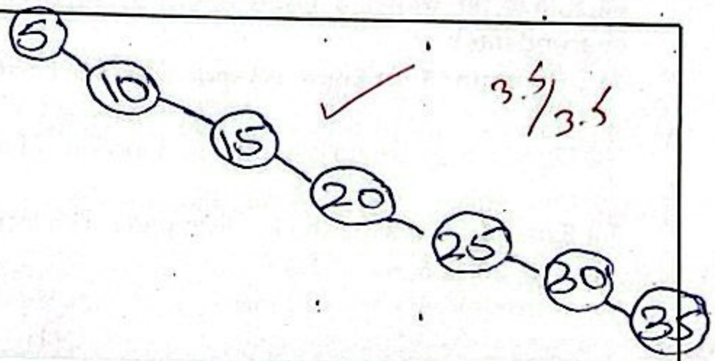
National University of Computer and Emerging Sciences (Lahore)			
Course:	Applied Programming	Code:	CS-0319
Section:	MSCS-2A	Semester:	Spring 2024
Time:	25 minutes	TotalMarks:	10
Date:	20 March 2024	Roll no:	24I-7807
Name:	MARYAM SALEEM		

Question#1: [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:



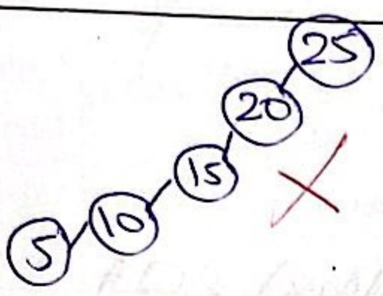
✓ 3.5/3.5

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:



~~0.5~~ 0.5

$\frac{}{1.5}$

Question#2:

[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:

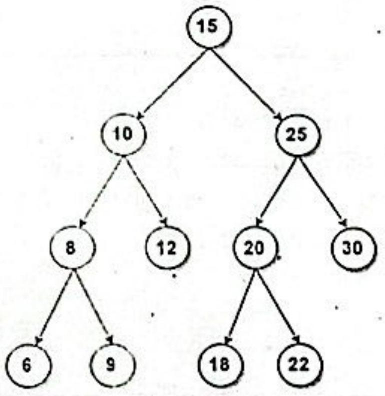
```
1)Node* insertIntoBST(int * arr, int size)
```

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.



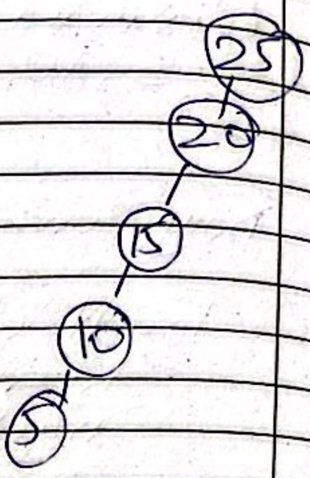
$\frac{0.5}{10}$

The subtrees are:

```
 6        9         8               10             12         18
                   / \             / \
                  6   9           8   12
                                 / \
                                6   9
```

i = 3 .

newRoot = 25
root->right = NULL
newRoot -> left = 20
return 25.



Question 2:

arr -> unsorted array.

```
Function insertR(int* arr, int size)
{
    sortedArray = mergesort (arr).
    for (int i=0; i<=size; i++){
      if (sortedArray[i] > node)
      {
         node -> left = sortedArray[i].
```

0.5/10

Scanned with CamScanner

Part No.

MARYAM SALEEM 241-7807

Rough Work

Question#1.

5, 10, 15, 20, 25, 30, 35.

4.5/15

Dry Run:

```
TreeNode* left ~~[illegible]~~ Rotate (TreeNode* root) {
    TreeNode* newRoot = root->right
    newRoot->left = root;
    return newRoot;
```

~~for~~ 5:

~~function insertR (Node* node, d) {~~

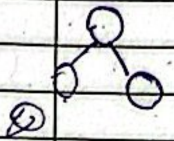
```
newRoot = 10
newNode->left = ~~[illegible]~~ 5
   return 10
```

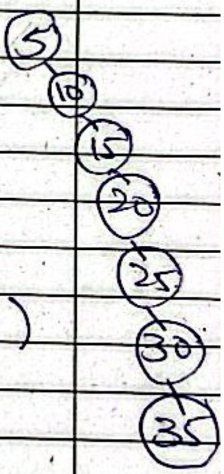
~~newNode =~~

```
TreeNode * RotationThrice (                )
  for i = 0 to 3 {
      i = 0
      newNode = 10
      NewNode -> left = NULL
         return 10
      i = 1
      newNode = 15
      ~~newNode -> left =~~
      root -> right = NULL
      newRoot -> ~~left~~ = 10

  i = 2

  newNode* newRoot = 20
  root -> right = NULL
  newRoot -> left = 15
       return 20
```




5
10
15
20
25
30
35