|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ***National University of Computer and Emerging Sciences, Lahore Campus*** | | | | |
| C:\Users\saif\AppData\Local\Microsoft\Windows\Temporary Internet Files\Content.Word\final design.jpg | **Course:** | Data Structures | **Course Code:** | CS 201 |
| **Program:** | BS(CS) | **Semester:** | Fall 2019 |
| **Duration:** | 15 Minutes | **Total Marks:** | 10 |
| **Paper Date:** | 24 Oct 2019 | **Exam** | Quiz 3 |
| **Section:** | **A** |  |  |
| **Instruction/Notes:** | Solve the exam on this question paper. | | | |

**Question:** Assume that you have a binary tree class. Implement a C++ program that prints the data of the binary tree node if it has same number of nodes in its right and left sub-trees. The program should also return the total of such nodes in the current binary tree.

|  |  |
| --- | --- |
| **Input**  (Diagram) | **Output**  **2 11 22 28 25 15**  **and return**  **6** |

**int binarytree\_count\_recursive(const btnode \*root, int&coutCount)**

**{**

**int TotalCount = 1, RightCount=0, LeftCount=0 ;** // count self ---------- 1 marks

**if (root->left != NULL)** // count left ---------- 2 marks

**{**

**RightCount = binarytree\_count\_recursive(root->left);**

**}**

**if (root->right != NULL)**

**{** // count right ---------- 2 marks

**LeftCount = binarytree\_count\_recursive(root->right);**

**}**

**if (RightCount==LeftCount)** // node count compare ---------- 1 marks

**{**

**cout<<root -> data;** // cout data ---------- 1 marks

**coutCount++;** // count couts ---------- 1 marks

**}**

**TotalCount+= (LeftCount+ RightCount);** // count merge ---------- 2 marks

**return TotalCount;**

**}**

**Int program()**

**{**

**int count=0;**

**if(root!= null)**

**binarytree\_count\_recursive(root, count)**

**return count;**

**}**