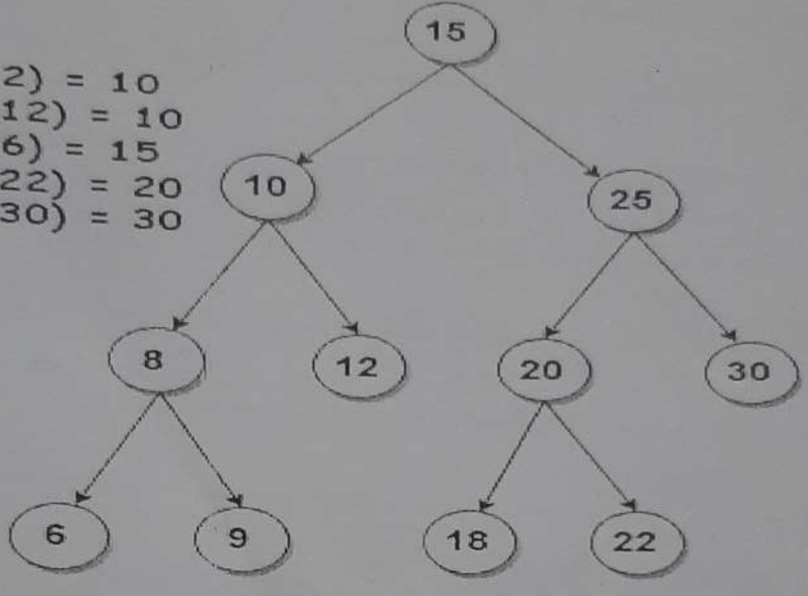
National University of Computer and Emerging Sciences, Lahore Campus

Course:	Data Structures Lab	Course Code:	CL218
Program:	BS(Computer Science)	Semester:	Fall 2019
Duration:	15 Minutes	Total Marks:	10
Paper Date:	05-Nov-2019	Weight	5%
Section:	B	Page(s):	1
Exam:	Quiz-1	Roll No:	[redacted]

LCA (6, 12) = 10
LCA (10, 12) = 10
LCA (20, 6) = 15
LCA (18, 22) = 20
LCA (30, 30) = 30



Lowest Common Ancestor in BST

Let T be a rooted tree. The lowest common ancestor between two nodes n1 and n2 is defined as the lowest node (farthest from the root node) in T that has both n1 and n2 as descendants (where we allow a node to be a descendant of itself). Write a function LCA which returns the value of the node that is the lowest common ancestor of two given keys. Assume that the two keys always exist in the BST. Try to write an efficient algorithm.

```
Struct TNode
{
 int key;
 TNode *lChild, *rChild;
};
int LCA( TNode *root, int k1, int k2)
```