National University

Of Computer & Emerging Sciences Faisalabad-Chiniot Campus

Question 1: (03) Marks 05

What does the following method do, describe briefly? Trace the code for several different values to prove your answer. No points will be given for direct answer.

```
BSTnode* guess(BSTnode* root, int X)
{
  if (root == null)
    return null;
  if (root→data > X)
  {
    BSTnode* temp = guess(root→left, X);
        if (temp == null)
          return root;
        else
          return temp;
  }
  else
      return guess(root→right, X);
}
```

01 For search guess

answer:
it gives the address of the value having value equal or above :7
our node is in tree, till it found Null.
:7 tree Node is not in t
:t return the address node one above :t

Answer: *write your final answer after tracing the given code for several different values*

→ if for that the Node we are searching's right node is empty, it gives that the address to that Node.

→ otherwise gives the value

Finally node of value 18 is returned

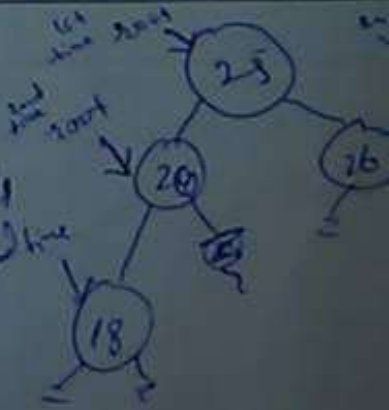
guess(root, 18) ← return node of value 18

temP = guess(root→left, 18) // :7 (root→data > x)

temP = guess(root→left, 18) return node 18

guess(root→right, 18) Null

//else because 18 is not true



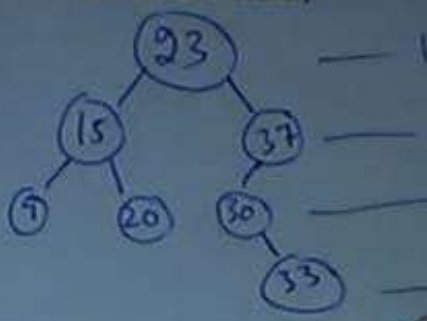
for guess(root, 20) ← Node of value 20.

temP = guess(root→left, 20) ← node of value 20

Question 2: **Marks 0.5+0.5+0.5+0.5+0.5+1+1 = 4.5**

Consider a BST which is constructed by following node values (in the given order): 23, 15, 37, 9, 20, 30, 33. Draw the BST and answer the following questions.

(DRAW BST HERE)

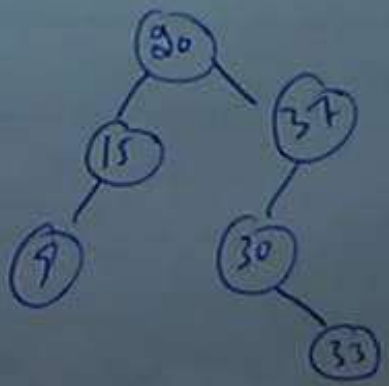


3.5

S #	Questions	Your answer(s)
i.	How many nodes are required to make it a full tree?	06 ✗
ii.	What is the maximum level number of the tree?	03 ✓
iii.	What nodes are on Level 3?	having value 33 ✓
iv.	Which node(s) has sibling?	15 and 37 are siblings. 9 and 20 are siblings
v.	Trace the shortest path of the tree.	Given tree is not's shortest Path. ✗

vi. Delete 23, draw new BST (use the BST drawn from given data set).

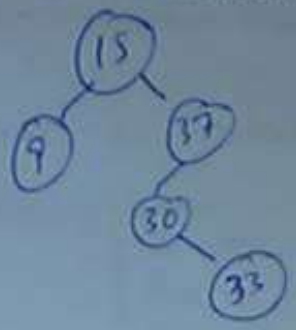
(DRAW BST HERE)



✓

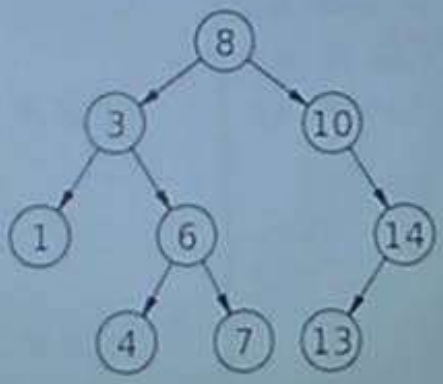
vii. Delete 20, draw new BST (this time use BST drawn after deletion of 23).

(DRAW BST HERE)



Question 3: **Marks 02+1.5 = 3.5**

Show how the following BST will be represented sequentially in the memory such that left and right child of any 'k' node can be accessed directly *(as discussed in the lecture)*.



Solution:

using Array.

National University

Of Computer & Emerging Sciences Faisalabad-Chiniot Campus

Question 4:	Good (05)	Marks 05

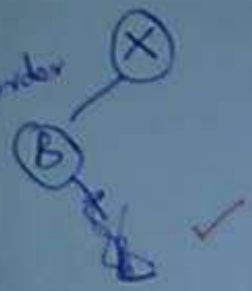
You are given two traversal orders namely preorder and inorder respectively of same binary tree. Can we construct a **unique** binary tree from these? Answer YES or NO. If your answer is YES then draw the UNIQUE binary tree and show complete working (draw diagrams separately after each step). And if your answer is NO then provide at least two binary trees which may be constructed from the given two orders. No points will be given for partial and/or incomplete answer.

Preorder: *X B L K E W G H P* | **Inorder:** *B L X E K G H W P*

Solution: *draw diagrams in both portions. Use back side, if required*

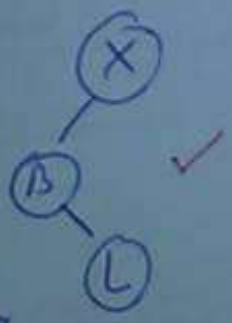
Yes

⇒ First Node in binary tree Preorder is root.



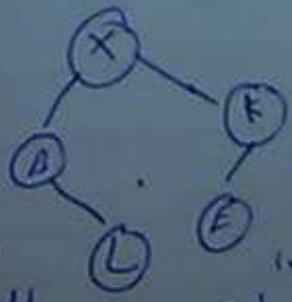
X B ✓

2nd step.
From inorder X says B is in in it's left and B says L is in right



X B L ✓

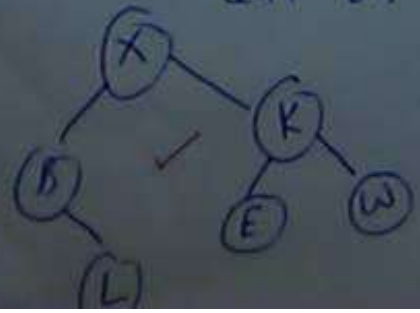
3rd Step:
Inorder is LNR, so after root X there are all nodes are on right.



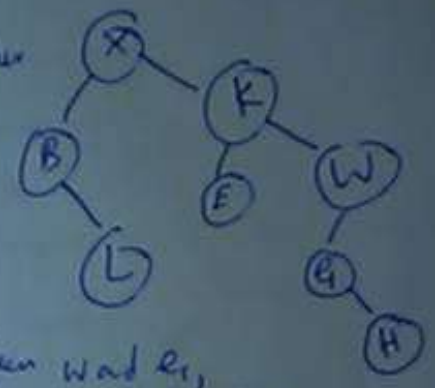
X B L K E

Pre order say K is First then E and In order says K is in right of X. and E is in left of K.

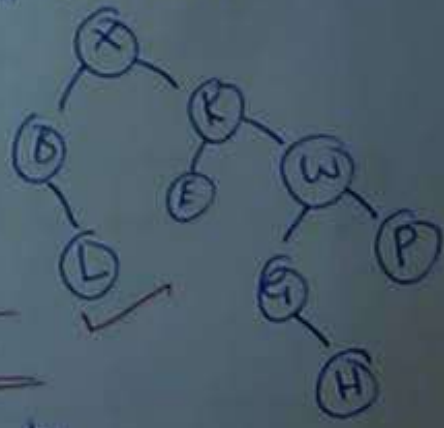
4th Step
After in sequence of Pre order there



5th Step.
Now in Preorder there is G, and In order say it's in left of W and there is only H between W and G, so it must be in right of G



2.5



2.5

② According to inorder rule P is in right of W.

National University

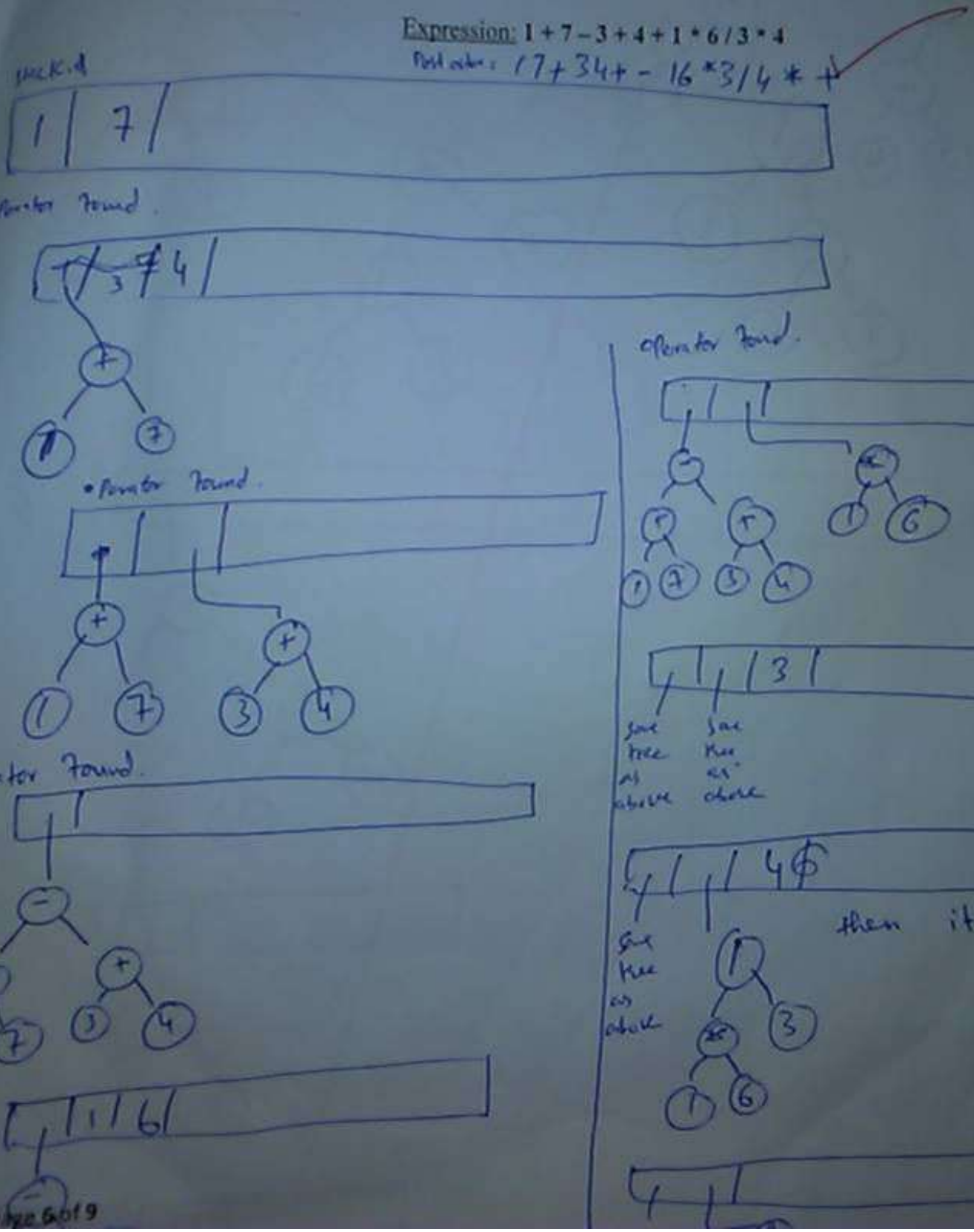
Of Computer & Emerging Sciences Faisalabad-Chiniot Campus

Question 5: (0.25 effort) **Marks 02+02 = 04**

a. Draw expression tree for the following arithmetic expression. Draw diagram after each step clearly. No points will be given for direct answer.

Expression: 1+7-3+4+1*6/3*4

Postfix: 17+34+-16*3/4*+

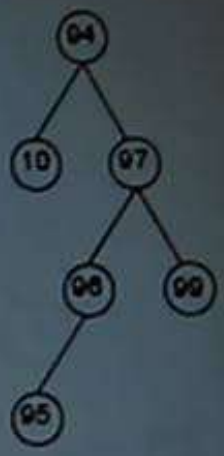


Question 6: **Marks 05+01 = 06**

a. Implement the following non-recursive function *levelTraversal* which displays (integer) nodes of a binary tree at each level from left-to-right. You may use any other data structure *(if required, no need to implement basic operations)* other than an array and a linked list.

void levelTraversal (TreeNode *root);

For example: 94, 10, 97, 96, 99, 95 is the *levelTraversal* sequence for the given binary tree.

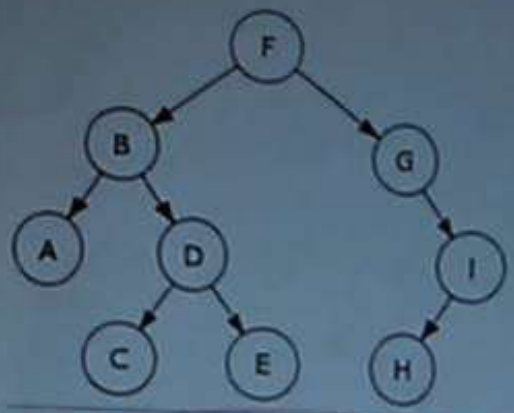


Solution:

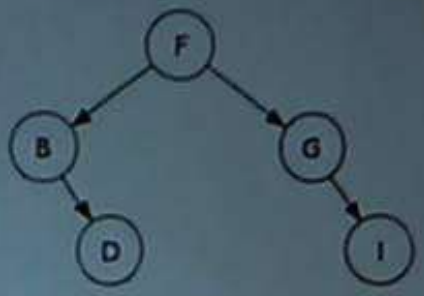
b. What is the complexity of your code?

Question 7: Marks 06

Write a recursive function **removeLeaves** that removes all the current leaves from the binary tree. You cannot use static variables and/or pointers for any purpose. Note that the new leaves of the tree will be the nodes that just had (old) leaves as children. For example:



Original Tree



Tree after removal of leaf nodes

Solution:

To Destroy the Complete tree:

```
void removeLeaves(Node *current)
{
    if (curr != Null)
    {
        removeLeaves(current->left);
        removeLeaves(current->right);
        Node *temp = current;
        delete temp;
    }
}
```

0.5

To remove only leaf nodes:

```
void removeLeaves(Node *current)
{
    if (curr != Null)
    {
        if (curr->left != Null && curr->right != Null)
        Node *temp = current;
        delete temp;
    }
}
```