

Question 2 Recheck
 Algo is correct
 #

In Q5 I just forgot
 to write (2)
 is part marking
 not valid in this
 question

22 + 2 = 24
 Show paper
 directly to
 + 2 should
 be given

CSL201 Data Structures

Minor 1

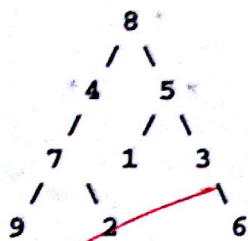
30th Aug, 2013

Name ~~XXXXXXXXXX~~ Entry Number ~~XXXXXXXXXX~~ Gp No. 05

Your Lab day Friday Your TA Divyanshu

Answer all the questions in the space provided for each question. You may use the answer script for rough work.

1. [2] Write the preorder traversal of the following tree:



8 4 7 9 1 2 5 1 3 6

8 4 7 9 2 5 1 3 6

2. [9] Complete the following function which accepts a sorted linked list in ascending order, with each node containing an integer, and returns the list after removing all nodes which appear more than once. Thus given the input

34 - 40 - 55 - 55 - 62 - 78, it should return 34 - 40 - 62 - 78

node * duplicate(node *A) {

node * curr;

curr = A;

while (curr -> next != NULL)

if (curr -> data != curr -> next -> data) { curr = curr -> next;

else {

node * temp;

temp = curr -> next;

curr -> next = temp -> next;

delete (temp);

~~curr = curr -> next;~~

}
 return A;

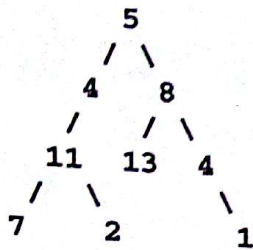
}

1

Have's b
 removed first
 repeated
 entry.

~~you are not
 checking for
 multiple
 occurrence~~

3. [10] Given a tree and an integer, write a function which returns true if there is a path from the root down to a leaf, such that product of all the values along the path equals the given integer, otherwise return false. For example given the tree



Root-to-leaf paths for the above tree :

- path 1: 5 4 11 7
- path 2: 5 4 11 2
- path 3: 5 8 13
- path 4: 5 8 4 1

For this problem, we will be concerned with the product of the values of such a path -- for example, the product of the values on the 5-4-11-7 path is $5 * 4 * 11 * 7 = 1540$.

```

void
void check (node *p, int d) {
int counter = p->data;
switch:
case
  
```

```

check (node *p, int d) {
  if (d % (p->data) != 0)
    return false;
  else {
    switch:
      case 1: check (p->left, d/(p->data))
  }
}
  
```

look for

```

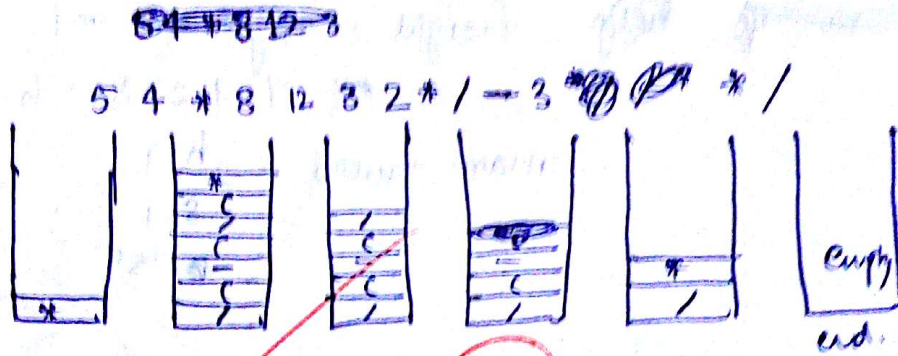
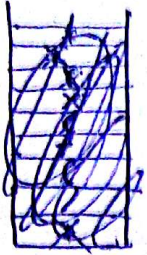
check (node *p, int d) {
  int p = d % p->data;
  int q = d / p->data;
  if (p == 0) return false;
  else if (p->left == NULL)
    if (q / p->left->data == 1 && q % p->left->data == 0)
      return true;
  else
  
```

no before

2

4. [6] Convert the following expression into postfix notation using a stack. Show the entries in the stack every time a character is read from the expression.

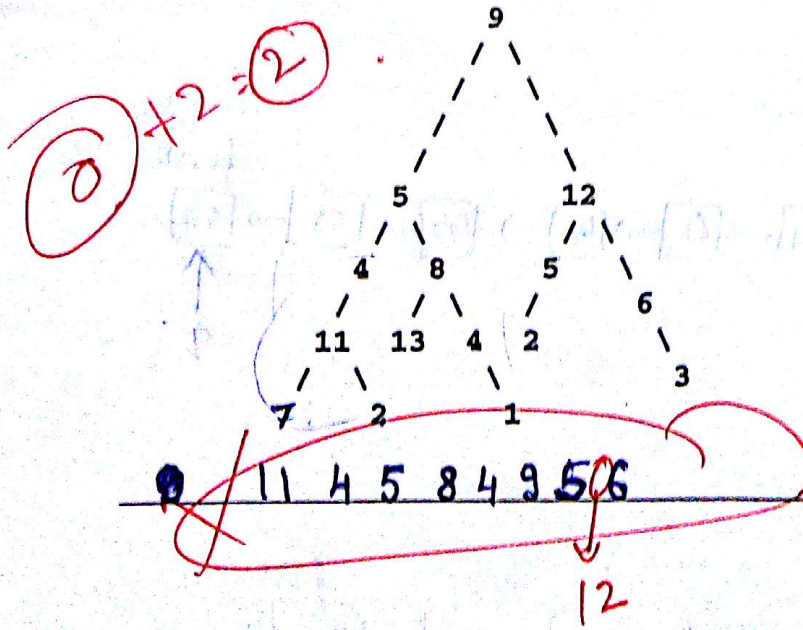
$5 * 4 / (8 - (12 / (3 * 2))) * 3$



6

5. [5] What will be the output of the following function if it takes the given tree as input ?

```
void wonder ( node *p) {
    wonder ( p -> left);
    if (p -> right != NULL || p -> left != NULL){
        cout << p -> data << " ";
    }
    wonder (p -> right);
}
```



6. [3] What is the maximum number of nodes in a binary tree if it contains 120 leaf nodes? Justify your answer.

~~leaf node~~ \rightarrow ~~excludes the root node and external node~~

~~total no. of leaf~~ height : $\lg(120) + 1$
 $\approx 7 + 1 = 8 = h$

0

~~max. nodes~~ $= 2^h - 1$
 $= 2^8 - 1$
 $= 255$

leaf nodes will be more than 120 in this case.

8. If you are assuming complete binary tree, the answer should have been 239 (114 + 120), but the correct answer is INFINITE.

7. [5] Indicate how the list below will change when we call the following routine
 $first \rightarrow 57 - 55 - 92 - 40 - 62 - 84 - 78$

```
void magic(node *first)
{
    struct node *p = first,
                *q = NULL,
                *r;
    while (p != NULL)
    {
        r = q;
        q = p;
        p = p->next;
        q->next = r;
    }
    q = first;
}
```

8

