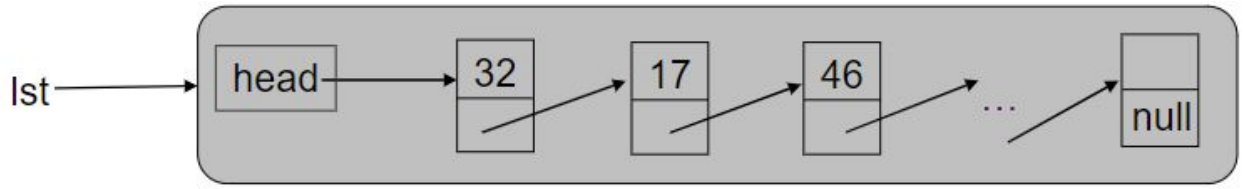
EECS 233 Written Assignment #2
100 points
Due October 12, 2018 before 11:59pm

Submission Guidelines
Submit all your solutions in the form of a PDF file to Canvas. You can either type them up or write on a piece of paper and scan them. If you choose the latter, please make sure that your hand writing is clear and readable. Name your file as W2_YourCaseID_YourLastName.pdf

1. Let $f(n) = O(g(n))$ and $t(n) = O(g(n))$. Prove or disprove that $f(n) = \Omega(t(n))$.

2. Consider a singly linked list.



Write a linear-running time method `public void reverse()` that would be invoked on a list object (e.g., lst.reverse()) and would reverse the list using only constant space (note that this precludes using recursion since this would use a nonconstant space in the call stack memory).

3. Using Stacks(`java.util.Stack<E>`), write a program that evaluates a prefix arithmetic expression with **single-digit** integer operands and operators '-', '+', '*', and '/'. For simplicity, assume that your input string representing the expression does not have any errors (e.g., is valid – no error detection is needed), and that it contains only digits and operator characters (e.g., no spaces).
Hint: You might want to use the string library method charAt(int n).

```
public static int postfixExpr (String expr) {
      ……
}
```

4. Consider a perfectly balanced (also called simply "perfect") binary tree, with all leafs being at the same depth from the root. Let the number of nodes be n.
    a. What's the height of this tree?
    b. What's the number of leaf nodes in such a tree?
    c. What's the number of internal (non-leaf) nodes? The ratio of the number of leaf to non-leaf nodes?
    d. If the tree has n nodes, for which n is it possible to arrange nodes into a perfect binary tree?

5. Write a recursive implementation of insertion into binary search tree.

```
public void insert(Node root, Node new) {
      ……
}
```

6. This question has three subparts which follow one another.
    a. Insert 48, 61, 53, 33, 70, 65, 27, 77, 42 to an empty AVL tree, in the given order. Show only the final tree after all insertions.
    b. Add one more element such that it causes a single right rotation in the tree. State the added number and show the final tree after the insertion.
    c. Insert 29 to the AVL tree. Now, delete one element such that it causes a single left rotation in the tree. State the deleted number and show the final tree after the deletion operation.

7. Does the insertion order of the same set of elements into an AVL tree affects the resulting tree structure? If yes, provide an example; if no, why.