CSE 101
Midterm 2 Review Problems

1. Use limits to show that the following statements are true.

   a. $n^2 = \omega\left(n\sqrt{n}\right)$

   b. $2^n = o(3^n)$

   c. $3^{(2^n)} = o\left(2^{(3^n)}\right)$ Hint: use the result of part (b)

   d. If $h(n) = o\left(f(n)\right)$, then $f(n) + h(n) = \Theta\left(f(n)\right)$

2. Rank the following functions from lowest to highest asymptotic growth rate.

   1) 2^n
   2) $n \ln(n)$
   3) n
   4) $2^{\ln(n)}$
   5) ln(ln(n))
   6) $n\sqrt{n}$
   7) n^2
   8) $\ln(n^2)$
   9) $\sqrt{n}$

3. Consider the integer List ADT from pa5. Write a C++ client function with the heading

   ```
   List Replace(List L, int x, int y)
   ```

   that returns a new List containing the same elements as L, in the same order, but with all occurances of the target x replaced by y. For instance, if $L = (3, 2, 9, 2, 5, 7, 2, 6, 4, 2, 7, 8, 2, 9)$, $x = 2$ and $y = 1$, then the returned List would be $(3, 1, 9, 1, 5, 7, 1, 6, 4, 1, 7, 8, 1, 9)$. The returned List will have its cursor placed at its back. If L does not contain the element x, then the returned List will be a copy of L. This function has no preconditions.

4. Consider the integer List ADT from pa5 again. Let us say that two Lists A and B are *similar* iff they contain the same elements with the same frequencies (though possibly in different orders). For instance, the Lists $A = (1, 2, 1, 1, 2)$ and $B = (2, 1, 2, 1, 1)$ are similar, since they both contain three $1's$ and two $2's$. On the other hand, Lists $A = (1, 2, 1, 1, 2)$ and $C = (1, 1, 2, 1, 1)$ are not similar since C contains four $1's$ and one 2. Note that similar Lists necessarily have the same length, and that two empty Lists are deemed to be similar. Write a C++ client function with heading

```
bool Similar(List A, List B)
```

that returns `true` if A and B are similar Lists, and `false` otherwise. This function has no preconditions.

5. Suppose we alter the List ADT from pa5 by doing

```
typedef char ListElement;
```

at the beginning of `List.h`, making it a list of `char` instead of `int`. Assume a List L consists entirely of parenthesis characters `'('` and `')'`. The List L is called a *Well Formed Formula* (WFF) iff all parentheses can be matched in pairs (open and close). For instance "`(()(()))`" and "`()()(())`" are WFFs, while "`(()()`" and "`(()))`" are not. The empty List is considered to be a WFF. Write a client function with heading

```
bool isWFF(List L)
```

that returns `true` or `false`, according to whether `L` is or is not a WFF. (Hint: search for adjacent matching pairs and delete them. If `L` becomes empty, then return true.)

6. Let T be a BST containing the keys $\{1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14\}$. Suppose that a **post-order tree walk** prints the keys in order: $1, 3, 5, 4, 2, 7, 9, 10, 12, 11, 14, 13, 8, 6$. Determine the structure of T. Present your solution either by drawing a picture of the tree, or by constructing a table giving the parent of each Node.

7. Use the `TreeInsert()` algorithm to insert the following keys: 9, 2, 1, 5, 3, 4, 7, 6, 8, 14, 10, 12, 11, 13, 15 (in order) into an initially empty BST.

   a. Draw the resulting BST.

   b. Use the `Delete()` algorithm to delete the following keys: 9, 5, 2 (in order) from the BST you drew in part (a), then draw the resulting tree.