Exam 2 Review/Practice Questions

1) Consider the following arithmetic expression:

\[(4 - 1) * (20 - 10) - 10\]

a. Rewrite the expression above using postfix notation
b. Using the peek(), pop(), and push() methods, show how a stack can be used to support the evaluation of the expression in its postfix representation.

2) Consider the following method:

```java
public static boolean p(String s, int i, int f)
    if (i<f)
        if (s.charAt(i) == s.charAt(f))
            return p(s,i+1,f-1);
        else
            return false;
    else
        return true;
}
```

What does p(s,0,s.length-1) return
a) if s ="UTEPE" 
b) if s ="SAMS' 
c) if s ="kayak" 
d) if s ="ABBA"

3) Write a recursive method that receives an integer n and prints all the integers from n to 0 in descending order

4) Write a recursive method that receives an integer n and prints all the integers from 0 to n in ascending order

5) Write a recursive method that receives an integer n>0 and prints the names of the digits in n.

6) Write a recursive method to multiply two integers a and b using addition:

\[a*b = 0 \text{ if } b ==0 \text{ and } a*b = a + a*(b-1) \text{ if } b>0 \text{ and } a*b = a*(b+1) -a \text{ if } b<0\]

7) Write a recursive method that receives a float x and an integer n and computes x^n using multiplication:

\[x^n = 1 \text{ if } n ==0 \text{ and } x^n = x*x^{(n-1)} \text{ if } n>0 \text{ and } x^n = x^{(n+1)}/x \text{ if } n<0\]
8) Consider your method from question 4. Which of the following two inputs will result in faster execution time?
   • a = 10000, b = 3
   • a = 3, b = 10000
   Modify your method from question 4 to enable it to run as quickly as possible.

9) Consider a stack implemented with a reference-based list. Show the configuration of the list representing the stack after each push and pop operation in the following code.

   ```java
   Stack S = new Stack();
   for (i=0; i<5; i++) {
     S.push(i);
     S.push(S.pop()+1);
   }
   ```

10) Consider the list 9, 23, 1, 6, 8, 12, 7, 18. Show how each of the sorting algorithms (bubblesort, selection sort, insertion sort, mergesort, and quicksort) will sort this list.

11) Describe the worst-case complexity for each of the sorting algorithms, and describe your reasoning for why this is the complexity.

12) Write a method that uses a stack to determine whether a given input string is a palindrome or not.

13) Write a recursive method for performing binary search.

14) Write a recursive algorithm for computing the Fibonacci numbers. Trace your code for computing the 5th Fibonacci number.

15) Write both a recursive and iterative method for computing n!.

16) For each of the following equations in infix notation, rewrite them in an equivalent postfix expression.
   a) a + b + c
   b) a / (b + c)
   c) a / b + c
   d) a * (b / c) + d
   e) (a * b + c) + f - g * h
   f) a - b * h / g + n * (x - y)