

Total Value: 50 points

Due Date: September 14, 2014 (1800)

1. Write a C++-style function or pseudocode that recursively calculates the factorial of a given integer. You must identify your base case and guardians. (10 points)

factor(8) = 8 * factor(7)
 factor(7) = 7 * factor(6)
 factor(6) = 6 * factor(5)
 factor(5) = 5 * factor(4)
 factor(4) = 4 * factor(3)
 factor(3) = 3 * factor(2)
 factor(2) = 2 * factor(1)
 factor(1) = 1 * factor(0)
 factor(0) = 1

```

int factor(int i) {
    if (i < 0) return; // Guardian, Sanitizer negative input
    else if (i > 0)
        return i * factor(i-1);
    else if (i == 0) // Base Case, when/if i == 0 returns 1
        return 1;
}
  
```

2. Write a snippet of C++ or pseudocode that computes the factorial iteratively (that is, in a simple loop). (10 points)

```

int factor(int x) {
    if (x < 0) throw (too_small);
    else if (x > 1) {
        for (i = x; i > 1; i--) {
            i *= (i-1);
        }
        return i;
    } else {
        return 1;
    }
}
  
```

3. For each of the following $f(n)$ functions, indicate whether or not ("Yes" or "No") they are bounded by the listed "Big- O " expressions. Also write down the tightest bound (which may not be in the table). (1 point each)

$f(n)$	Tightest upper bound	$O(n)?$	$O(n^2)?$	$O(1)$	$O(2^n)$
$n+3$	$O(n)$	Yes	Yes	No	Yes
$100n+2n^3$	$O(n^3)$	No	No	No	Yes
$5.5n^2+3n+1000$	$O(n^2)$	No	Yes	No	Yes
17	$O(1)$	Yes	Yes	Yes	Yes
$.0003n^5+50,000,001n+42$	$O(n^5)$	No	No	No	Yes
n^n	$O(n^n)$	No	No	No	No