Name: Max Copeland Date: 9/14/17

Pledge: "I pledge my honor that I have abided by the Stevens Honor System"

Give the complexity of the following functions. Choose the most appropriate notation from among O, θ, and Ω.

1.
```
void function1(int n) {
    for (int i = 1; i <= n; i++) {                  //n
       for (int j = i; j <= n; j += 2) {            //n
            cout << "*";                            //c
        }
    }
}
```

Answer: $\theta(n^2)$

2.
```
void function2(int n) {
    int count = 0;                                  //c
    for (int i = 1; i * i <= n; i++) {              //√n
        count++;                                    //c
    }
    cout << count;                                  //c
}
```

Answer: $\theta(\sqrt{n})$

3.
```
void function3(int n) {
    int count = 0;                                  //c
    for (int i = n/2; i <= n; i++) {                //n
        for (int j = 1; j + n/2 <= n; j++) {        //n
            for (int k = 1; k <= n; k *= 2) {       //lg(n)
                count++;                            //c
            }
        }
    }
    cout << count;                                  //c
}
```

Answer: $\theta(n^2 lg(n))$

4.
```
void function4(int n) {
    int count = 0;                                  //c
    for (int i = n/2; i <= n; i++) {                //n
        for (int j = 1; j <= n; j *= 2) {           //lg(n)
```

```
            for (int k = 1; k <= n; k *= 2) {       //lg(n)
                count++;                            //c
            }
        }
    }
    cout << count;                                  //c
}
```

Answer: $\theta(n\lg(n)^2)$

5.
```
void function5(int n) {

    if (n % 2 == 0) {
        return;                                     //c
    }
    for (int i = 1; i <= n; i++) {                  //n
        for (int j = 1; j <= n; j++) {              //c
            cout << "*";                            //c
            break;                                  //c
        }
    }
}
```

Answer: O(n)

6.
```
void function6(int n) {

    int count = 0;                                  //c
    for (int i = 1; i <= n/2; i++) {                //n
        for (int j = 1; j <= n/3; j++) {            //n
            for (int k = 1; k <= n/4; k++) {        //n
                count++;
            }
        }
    }
    cout << count;
}
```

Answer: $\theta(n^3)$

7.
```
void function7(int n) {

    for (int i = 1; i <= n; i++) {                  //n
        for (int j = 1; j <= n; j += i) {           //log(n)
            cout << "*";                            //c
        }
    }
}
```

Answer: $\theta(n\log(n))$

8.
```
void function8(int n) {
    int i = 1, s = 1;                    //c
    while (s <= n) {                     //√n
        i++;                             //c
        s += i;                          //c
        cout << "*";                     //c
    }
}
```

Answer: $\theta(\sqrt{n})$

9. Processing Arrays

   a. Suppose you have an unsorted array of integers of length n and want to sum all the elements inside it. What is the running time of your algorithm? θ**(n)**

   b. Suppose you have an unsorted array of integers of length n and want to determine if all the values inside are positive. What is the running time of your algorithm? **O(n)**

   c. Suppose you have a sorted array of integers of length n and want to determine the median value. What is the running time of your algorithm? θ**(1)**

10. $f(n) = 3n^2 + 4n + 2 \quad \in \theta(n^2)$ **True**

If true, prove it by giving *integral* values for the required constants c_1, c_2, and n_0. Choose the tightest values possible for the c_1 and c_2 constants. If false, show the contradiction.

$c_1 n^2 \leq 3n^2 + 4n + 2$

$c_1 = 3$ for all values $n \geq 1$

$3n^2 + 4n + 2 \leq c_2 n^2$

$c_2 = 4$ for all values $n \geq 5$

$n_0 = 5$