1. Problem 0.1 on page 8 of the textbook. In each of the following situations, write $O$, $\Omega$, $\Theta$ in the blank.

   (a) $n - 100 = \Theta(n - 200)$
   
   (b) $n^{1/2} = \Theta(n^{2/3})$
   
   (c) $100n + \log n = \Theta(n + \log^2 n)$
   
   (d) $n \log n = \Theta(10n \log(10n))$
   
   (e) $\log(2n) = \Theta(\log(3n))$
   
   (f) $10 \log n = \Theta(\log(n^2))$
   
   (g) $n^{1.01} = \Theta(n \log^2 n)$
   
   (h) $n^2 / \log n = \Theta(n \log^2 n)$
   
   (i) $n^{0.1} = \Theta(\log^2 n)$
   
   (j) $(\log n) \log n = \Theta(n / \log n)$
   
   (k) $\sqrt{n} = \Theta(\log^3 n)$
   
   (l) $n^{1/2} = \Theta(5^{\log_2 n})$
   
   (m) $n2^n = \Theta(3^n)$
   
   (n) $2^n = \Theta(2^{n+1})$
   
   (o) $n! = \Theta(2^n)$
   
   (p) $\log n^{\log n} = \Theta(2^{(\log_2 n)^2})$
   
   (q) $\sum_{i=1}^{n} i^k = \Theta(n^{k+1})$
2. Work problem 0.3(c) on page 9 of the textbook.

3. Consider the following C++ program.

   ```cpp
   void process(int n)
   {
     cout << n << " " << n%2 << endl;
     if(n > 1) process(n/2);
   }

   int main()
   {
     int n;
     cin >> n;
     process(n);
   }
   ```

   For example, if you enter the value 37, the output will look like this:

   ```
   37 1
   18 0
   9 1
   4 0
   2 0
   1 1
   ```

   For any positive integer input, say \( n \), the second column is a string of bits. What does that bitstring represent?
4. Each of these code fragments takes $O(n \log n)$ time, but not necessarily $\Theta(n \log n)$. Give the asymptotic complexity of each in terms of $n$, using $\Theta$ in each case.

(a) 
```cpp
for(int i = 1; i < n; i++)
    for(int j = 1; j < i; j = 2*j);
    cout << "Hello" << endl;
```

(b) 
```cpp
for(int i = 1; i < n; i++)
    for(int j = i; j < n; j = 2*j);
    cout << "Hello" << endl;
```

(c) 
```cpp
for(int i = 1; i < n; i=2*i)
    for(int j = 1; j < i; j++);
    cout << "Hello" << endl;
```

(d) 
```cpp
for(int i = 1; i < n; i=2*i)
    for(int j = i; j < n; j++);
    cout << "Hello" << endl;
```

(e) 
```cpp
for(int i = n; i > 1; i=i/2)
    for(int j = i; j > 1; j--);
    cout << "Hello" << endl;
```

(f) 
```cpp
for(int i = n; i > 1; i=i/2)
    for(int j = n; j > i; j--);
    cout << "Hello" << endl;
```

5. These problems are harder than the ones above. Given the asymptotic complexity of each fragment in terms of $n$, using $\Theta$.

(g) 
```cpp
for(int i = 1; i < n; i=2*i)
    for(int j = 1; j < i; j=2*j);
    cout << "Hello" << endl;
```

Hint: Use substitution. Let $m = \log n$, $k = \log i$, $l = \log j$.

(h) 
```cpp
for(int i = 2; i < n; i=i*i)
    cout << "Hello" << endl;
```

Hint: Use substitution. Let $m = \log n$, $k = \log i$.

(i) 
```cpp
for(int i = 2; i < n; i=i*i)
    for(int j = 1; j < i; j = 2*j)
    cout << "Hello" << endl;
```

Hint: Use substitution. Let $m = \log n$, $k = \log i$, $l = \log j$.

(j) 
```cpp
for(int i = n; i > 1; i = log i)
    cout << "Hello" << endl;
```

Hint: The answer is a function you’ve possibly never heard of. That function is defined on page 136 of the textbook. I will simply tell you the answer, and you will need to remember it.