1. Find values for c and n₀ (according to the definition of $O(\cdot)$ ) for $f(n)$ is $O(g(n))$, where
   a. $f(n)=7n^2+3n$
   $g(n)=n^4$
   b. $f(n)=n+2\log n$
   $g(n)=n\log n$
   c. $f(n)=1000$
   $g(n)=3n^3$
   d. $f(n)=7n$
   $g(n)=n/10$

2. True or false?
   a. $f(n)$ is $\Theta(g(n))$ implies $f(n)$ is $O(g(n))$
   b. $f(n)$ is $\Theta(g(n))$ implies $g(n)$ is $\Theta(f(n))$
   c. $f(n)$ is $\Omega(g(n))$ implies $f(n)$ is $O(g(n))$

3. Find functions $f(n)$ and $g(n)$ such that $f(n)$ is $O(g(n))$ and the constant $c$ for the definition of $O(\cdot)$ must be $>1$. That is, find $f$ & $g$ such that $c$ must be greater than 1, as there is no sufficient $n₀$ when $c=1$.

4. Write the $O(\cdot)$ run-time of the functions with the following recurrence relations
   a. $T(n)=3+T(n-1)$, where $T(0)=1$
   b. $T(n)=3+T(n/2)$, where $T(1)=1$
   c. $T(n)=3+T(n-1)+T(n-1)$, where $T(0)=1$

5. What’s the $O(\cdot)$ run-time of this code fragment in terms of $n$:

   ```java
   int x=0;
   for(int i=n;i>=0;i--)
       if((i%3)==0) break;
       else x+=i;
   ```