Q1:

1. Is 2n+1 = O(2n ) ?        True

2. Is 22n = O(2n )  ?        False

3. Is (0.25)n = O(n) ?        True

Q2:

For every given f(n) and g(n) prove that f(n) = $\boldsymbol{\theta}$(g(n))

1. g(n) = $n^3$, f(n) = $3n^3 + n^2 + n$

   O< $\boldsymbol{\theta}$> $\boldsymbol{\Omega}$

   C1*g(n)< f(n)> C2*g(n)

   C1* $n^3$< $3n^3 + n^2 + n$ > C2* $n^3$

   By / $n^3$

   Let c1=2 & c2=4

   So f(n) = $\boldsymbol{\theta}$(g(n))

2. g(n) = $2^n$, f(n) = $2^{n+1}$

   O< $\boldsymbol{\theta}$> $\boldsymbol{\Omega}$

   C1*g(n)< f(n)> C2*g(n)

   C1* $2^n$ < $2^n$ *2> C2* $2^n$

   By / $2^n$

   Let c1=1 & c2=3

   So f(n) = $\boldsymbol{\theta}$(g(n))

3. g(n) = ln(n), f(n) = log(n) + log(log(n))

   O< $\boldsymbol{\theta}$> $\boldsymbol{\Omega}$

   C1*g(n)< f(n)> C2*g(n)

   C1* ln(n) < log(n) + log(log(n))> C2* ln(n)

   C1* ln(n) < ln(n) / ln(10) + ln(ln(n) / ln(10))/ln(10)> C2* ln(n)

   C1* ln(n) <(ln(n)+ln(ln(n))/ln(10)> C2* ln(n)

   By / ln(n)

   Let c1=1/ 2ln(2) & c2=2/ ln(2)

   So f(n) = $\boldsymbol{\theta}$(g(n))

Q3:
For every given f(n) and g(n) prove that f(n) = ***O***(g(n)) or f(n) = ***Ω***(g(n))

1. f(n) = $n^3$ , g(n) = $n^2$

   $n^3 > c * n^2$
   so $\Omega(n^2) = n^3$

2. f(n) = log(n), g(n) = $\log^2(n)$

   $\log(n) < c * \log^2(n)$

   so $O(\log^2(n)) = \log(n)$

Q4:
Prove that the running time of an algorithm is **θ**(g(n)) if and only if its worst-case running time is ***O***(g(n)) and its best-case running time is ***Ω***(g(n))

**θ**(g(n)):

$c1 \cdot g(n) \leq T(n) \leq c2 \cdot g(n)$

O(g(n)):

$T\ worst\ (n) \leq c1 \cdot g(n)$

So $T\ worst\ (n) \in O(g(n))$

Ω (g(n)):

$T\ best(n) \geq c2 \cdot g(n)$

So $T\ best\ (n) \in \Omega(g(n))$

Q5:
Prove that $\boldsymbol{O}(g(n)) \cap \boldsymbol{\omega}(g(n))$ is the empty set.

$f(n) \geq c1 * g(n)$

$f(n) < c2 \cdot g(n)$

$c1 \cdot g(n) \leq f(n) < c2 \cdot g(n)$

$\therefore f(n) \in o(g(n))$ & $f(n) \in \Omega(g(n))$

So $\Omega(g(n)) \cap o(g(n)) = \emptyset$