```
01. LEMA: existe m tal que, para todo n >= m, f(n) >= g(n),
    onde f(n) = n^2 = e g(n) = 20n.

===========================
---------- PROVA ----------
Definamos m = 21 e seja n >= m.
Temos então:
  f(n)  = n*n
       >= m*n (porque n >= m)
       >= 20n
        = g(n), CQD.
---------- PROVA ----------
===========================

02. DEFINIÇÃO: dadas duas funções f e g,
  com domínio igual aos naturais e contradomínio igual aos reais,
  nós dizemos que f = O(g) sse existem um natural n_0 e um real c, ambos
positivos,
  tais que, para todo n >= n_0, temos 0 <= f(n) <= c*g(n).

03. EXERCÍCIO: utilizando a definição acima, mostre que,
    para quaisquer constantes reais positivas a e b, a*n + b = O(n).

==============================
---------- RESPOSTA ----------
Sejam a e b constantes reais positivas quaisquer,
  e sejam f(n) = a*n + b e g(n) = n;
  temos que mostrar que f = O(g).
Sejam então c = a+1 e n_0 = b.
Logo, para todo n >= n_0, temos:
  c*g(n) = (a+1)*n
         = a*n + n
        >= a*n + b (porque n >= n_0 = b)
         = f(n)
        >= 0       (porque a e b são positivas e n é natural).
Logo, para todo n >= n_0, vale
  0 <= f(n) <= c*g(n), CQD.
---------- RESPOSTA ----------
==============================

04. O exercício acima ilustra a razão pela qual,
    ao analisarmos o tempo de execução de um algoritmo,
    nós geralmente descartamos as constantes envolvidas
    (por exemplo, o tempo constante utilizado na inicialização
     da variável contadora do algoritmo de busca linear,
     ou o tempo constante gasto pela instrução de retorno da função,
     após o laço do algoritmo).
Como vimos no exercício, tais constantes geralmente não influenciam
  a complexidade assintótica do tempo de execução dos algoritmos que tratamos.

05. EXERCÍCIO: mostre que n^3 != O(n^2).

==============================
---------- RESPOSTA ----------
Sejam f(n) = n^3 e g(n) = n^2; temos que mostrar que f != O(g).
Suponhamos, por absurdo, que f = O(g).
Logo, pela definição, existem um natural positivo n_0
  e um real positivo c tais que, para todo n >= n_0,
  0 <= f(n) <= c*g(n).
Sejam, então, tais c e n_0.
Seja, agora, n = máx {teto(c) + 1, n_0}; observe que n >= n_0.
Temos então:
  f(n) = n^3
       = n * n^2
```

      >= (teto(c) + 1) * n^2 (porque teto(c)+1 <= máx{teto(c)+1, n_0})
       > c * n^2
       = c * g(n), uma contradição com a nossa hipótese.
Logo, f != O(g), CQD.
---------- RESPOSTA ----------
==============================

06. DEFINIÇÃO: dadas duas funções f e g,
  com domínio igual aos naturais e contradomínio igual aos reais,
  f = Ômega(g) sse existem um natural n_0 e um real c, ambos positivos,
  tais que, para todo n >= n_0, 0 <= c*g(n) <= f(n).

07. EXEMPLO: mostre que n = Ômega (20*n).

==============================
---------- RESPOSTA ----------
Sejam f(n) = n e g(n) = 20*n; temos que mostrar que f = Ômega(g).
Sejam então c = 1/20 e n_0 = 1.
Assim, para qualquer n >= n_0, temos:
  c*g(n) = (1/20)*20*n
         = n
         = f(n).
Como c*g(n) >= 0 para todo n natural, então
  0 <= c*g(n) <= f(n) para todo n >= n_0,
  e portanto f = Ômega(g), CQD.
---------- RESPOSTA ----------
==============================