1.

   a. 2^(lg(n) + lg(n)) / n
      2^(2 * $\log_2$(n)) / n
      2^($\log_2$(n^2)) / n
      2^($\log_2$(n^2)) / n
      n^2 / n
      n
   b. Yes because 2 ^ lg(n) = 2 ^ $\log_2$(n) = n. so for n = 0 and c1 = 1, c1*2 ^ lg(n) = 2 ^ lg(n) and for c2 = 2, 2 ^ lg(n) < c2 * 2 ^ lg(n)
   c. Yes because 2 ^ (2 ^ (lg lg n)) = 2 ^ (2 ($\log_2$($\log_2$(n)))) = 2 ^ ($\log_2$(n)) = n. so just like 1b n = 0 and c1 = 1, c1*2 ^ lg(n) = 2 ^ lg(n) and for c2 = 2, 2 ^ lg(n) < c2 * 2 ^ lg(n)

2. Loop invariant = prod(i) + x(i) * y(i) = a*b. It holds for i because x and y both change but change in ways that makes them still equal to a * b. It will hold for i + 1 as well because x will be divided by 2 and y will be multiplied by 2 so it will always x(i) * y(i). x will eventually become 0 though and the loop will always conclude.
3. lg(n) << ln(n) << √(n) << n << nlogn << n^2 << 2^n << n^(1/n) << n^(1+1/lg(n)) << (lg(n))^1000 << 2^(√(lg(n))) << (√2)^(lg(n)) << (lg(n))^(√(2)) << n^(√(2)) << (1 + 1/n)^n << n^(1/1000) << H(n) << H(√(n)) << 2^(H(n)) << H(2n) << F(n) << F(n/2) << lg(F(n)) << F(lg(n))