1. The algorithm consists of two parts. The first part has two loops and the second part is executed if the condition is true.
   In the first part, the assignment operation inside the second loop is executed
   $$n + (n-1) + \cdots + 2 + 1 = \frac{n(n+1)}{2}$$
   times , and the if statement is executed n times. Therefore the complexity for the first part is
   $$T1(n) = n + \frac{n(n+1)}{2} = \frac{n^2}{2} + \frac{3n}{2}$$
   The second part consists of a nested loop structure. The first inner loop iterates exactly n^2 times. And the second inner loop is executed n*2^(n/2) times. Therefore the complexity for the second part is:
   $$T2(n) = n * 2^{n/2} + n^2$$

   Worst case complexity of the algorithm occurs when both parts of the function are executed. Thus the worst case complexity can be stated as:
   $$W(n) = T1(n) + T2(n) = \frac{n^2}{2} + \frac{3n}{2} + n * 2^{\frac{n}{2}} + n^2$$
   $$= \theta(n * 2^{\frac{n}{2}})$$

   In order to calculate the average complexity, we must understand when the second part is executed. Given a string of length n, the second part is not executed if any digit x[i] is not equal to the digit x[n-i]. That is to say the string must equal to itself written backwards. Such a string is called a palindrome. A palindrome is distinguished by the first half of it, therefore the number of palindromes of length n is equal to the number of strings of length n/2, which is:
   $$\#\ of\ palindromes\ of\ length\ n:\ 2^{\frac{n}{2}}$$
   $$\#\ of\ strings\ of\ length\ n: 2^n$$
   In the average case the first part of the algorithm will be executed and then the second part will be executed with the probability of
   $$P(n) = \frac{2^{\frac{n}{2}}}{2^n} = \frac{1}{2^{\frac{n}{2}}}$$
   Then the average time complexity is:
   $$A(n) = T1(n) + P(n) * T2(n)$$
   $$= \frac{n^2}{2} + \frac{3n}{2} + \frac{1}{2^{\frac{n}{2}}}\left(n * 2^{\frac{n}{2}} + n^2\right)$$
   $$\sim = \frac{n^2}{2} + \frac{5n}{2}\ for\ large\ n$$
   $$= \theta(n^2)$$
2. Consider the following functions:
   $$f(n) = \sin n + 1$$
   $$g(n) = \sin(n + \pi) + 1$$
   These functions are periodic and out-of-phase which means that there is no n_0 that statisfies:

$$c * f(n) \geq g(n), \qquad n \geq n_0$$
$$or: \quad c * g(n) \geq f(n), \qquad n \geq n_0$$

3. First we will rewrite the function in an open form using Stirling's approximation:

$$f(n) = n^4 \log(n!) + \sum_{i=1}^{5} i^n$$
$$n! \sim \sqrt{2\pi n}\frac{n^n}{e^n}$$
$$f(n) \sim n^5 \log(n) - n^4 \log(e) + \frac{1}{2}n^4 \log(n) + \frac{1}{2}n^4 \log(2\pi) + 5^n + 4^n + .. + 1$$

We can now answer the question:

a) False. Largest term of f(n), which is 5^n, grows exponentially faster than $n^5\log(n)$. Formally we cannot find constants to satisfy $c * n^5\log(n) \geq f(n)$

b) True. It is straightforward that $f(n) \geq n^5\log(n)$ for large n. However this is a trivial lower bound for the function.

c) True. Largest term of f(n) is 5^n. There are nine terms in the function, thus:

$$9 * 5^n \geq f(n) \geq 5^n, for\ large\ n$$

d) False.

$$o(5^n) = \left\{t(n) \mid \lim_{n\to\infty}\left(\frac{t(n)}{g(n)}\right)\right\} = 0$$

However:

$$\lim_{n\to\infty}\left(\frac{f(n)}{g(n)}\right) = \frac{n^5 \log(n) - n^4 \log(e) + \frac{1}{2}n^4 \log(n) + \frac{1}{2}n^4 \log(2\pi) + 5^n + 4^n + .. + 1}{5^n}$$
$$\lim_{n\to\infty}\left(\frac{f(n)}{g(n)}\right) = \frac{5^n}{5^n} = 1$$