The “master theorem” provides tight asymptotic bounds to recurrences of the following form (provided they meet some conditions):

$$T(n) = aT(n/b) + f(n).$$

The general conditions are that $a \geq 1$ and $b > 1$ are constants, and that $f(n)$ is an asymptotically positive function (that is, $f(n)$ is positive for some sufficiently large $n$).

The master theorem covers three cases:

1. If $f(n) = O(n^{\log_b a - \epsilon})$ for some constant $\epsilon > 0$, then $T(n) = \Theta(n^{\log_b a})$.
2. If $f(n) = \Theta(n^{\log_b a})$, then $T(n) = \Theta(n^{\log_b a} \lg n)$.
3. If $f(n) = \Omega(n^{\log_b a + \epsilon})$ for some constant $\epsilon > 0$, and if $af(n/b) \leq cf(n)$ for some constant $c < 1$ and all sufficiently large $n$ (the “regularity condition”), then $T(n) = \Theta(f(n))$.

1. For each recurrence, solve (find an asymptotic bound) using the “master theorem.” Indicate which case applies (or if not applicable, explain why none would apply) and also state the bound.

(a) $T(n) = 16T(n/4) + n$

**Solution:** We have $a = 16$, $b = 4$, $f(n) = n$, and $n^{\log_b a} = n^{\log_4 16} = n^2$. Since $f(n) = O(n^{\log_4 16 - \epsilon})$ where $1 \leq \epsilon < 2$, case 1 of the master theorem applies and $T(n) = \Theta(n^2)$.

(b) $T(n) = 3T(n/3) + n/2$

**Solution:** We have $a = 3$, $b = 3$, $f(n) = n/2$, and $n^{\log_b a} = n^{\log_3 3} = n$. Since $f(n) = \Theta(n^{\log_3 3})$, case 2 of the master theorem applies and $T(n) = \Theta(n \lg n)$.

(c) $T(n) = 0.5T(n/2) + 1/n$

**Solution:** The master theorem does not apply here because $a < 1$. 

Recall that mathematical induction is a technique for proving a certain statement or property about the set (or a subset) of natural numbers. To do so, first a basis case of the statement is proven: this is where we show that the statement is true for a particular value of \( n \). Then, the inductive step is proven: here we assume the statement is true for some arbitrary value of \( n \) (this is called the induction hypothesis) and use that fact to show that the statement is true for \( n + 1 \).

2. Pick ONE statement to prove using mathematical induction. Indicate which part (a or b) you wish to have graded. Only ONE will be graded.

   (a) Prove the following statement using induction (\( n \) is a natural number).

   \[
   \sum_{i=0}^{n} 2i = n(n + 1)
   \]

   **HINT:** If you are unfamiliar with \( \Sigma \) notation, the following statement is equivalent:

   \[
   0 + 2 + 4 + 6 + 8 + \cdots + 2(n - 2) + 2(n - 1) + 2n = n(n + 1).
   \]

   In other words, the statement you are being asked to prove is: “the sum of the first \( n \) even numbers is equal to \( n(n + 1) \).

   **Solution:** The base case is \( n = 0 \):

   \[
   \sum_{i=0}^{0} 2i = 0(0 + 1) \equiv \\
   0 = 0
   \]

   For the inductive step, we want to prove that

   \[
   \sum_{i=0}^{n+1} 2i = (n + 1) [(n + 1) + 1] \equiv \\
   2(n + 1) + \sum_{i=0}^{n} 2i = (n + 1)(n + 2)
   \]

   Note that we re-wrote the expression by “moving out” the last term of the sum. Now the expression contains the induction hypothesis, which we can substitute in (and then expand/rearrange terms on both sides):

   \[
   2(n + 1) + n(n + 1) = (n + 1)(n + 2) \equiv \\
   2(n + 1) + n(n + 1) = n^2 + 3n + 2 \equiv \\
   2n + 2 + n^2 + n = n^2 + 3n + 2 \equiv \\
   n^2 + 3n + 2 = n^2 + 3n + 2 \equiv \\
   \]

   (b) Prove the following statement using mathematical induction. Assume that \( n \in \mathbb{N} \) and \( n \geq 3 \).

   \[
   n^2 > 2n + 1
   \]

   **HINT:** You may find the following set of properties regarding inequalities useful. Assume that \( a, b, \) and \( c \) are natural numbers. The properties are shown using \( < \) but they also hold using \( >, \leq, \) and
\[ \geq. \]

if \( a < b \) then \( a + c < b + c \)

if \( a < b \) then \( a \cdot c < b \cdot c \)

if \( a < b \) and \( b < c \) then \( a < c \)

**Solution:** The base case is \( n = 3 \):

\[
3^2 > 2(3) + 1 \equiv \\
9 > 7 \\
\checkmark
\]

For the inductive step, we want to prove that

\[
(n + 1)^2 > 2(n + 1) + 1 \equiv \\
n^2 + 2n + 1 > 2n + 3.
\]

To do this, add two to both sides of the induction hypothesis:

\[
n^2 > 2n + 1 \equiv \\
n^2 + 2 > 2n + 3
\]

Since \( n^2 + 2n + 1 > n^2 + 2 \) (trivially) and \( n^2 + 2 > 2n + 3 \) (by the induction hypothesis), the proof is complete.
3. Write a function in C AND in Haskell that reverses a list of integers. Your C function should use linked lists.
4. Write a function in C AND in Haskell that, given a list of integers, extracts only the even integers. Your C function may use a linked list or array as input, but the output should be a linked list. Your function should not modify the input list. To test if an integer is even, you may use the modulus operator in C \( (n \mod 2 == 0) \) and the `even` function in Haskell.