1. Write a macro \texttt{HALF} representing the function $\lfloor n/2 \rfloor$ without using recursion or \texttt{Y}.

2. Write a macro \texttt{LOG2} representing the function $\lfloor \log_2 n \rfloor$ without using recursion or \texttt{Y}. You may use \texttt{HALF} and any other macro designed in class.

3. Consider the following transformation rule for lambda expressions:

   Find the leftmost outermost occurrence of \((E_1 (E_1 E_2))\) and rewrite the expression as \((\bar{2} E_1) E_2\). (The check of equality of the two instances of \(E_1\) is purely syntactic. It is not an equivalence test.)

   (a) Suppose we started with \(E\), and used the above rule to get \(E'\). Is the normal form of \(E\) equivalent to the normal form of \(E'\), provided both exist? Prove or disprove.

   (b) Apply the above transformation to \(\bar{7}\) as many times as possible and show the steps to the final result, which we call \(\bar{7}'\).

   (c) In general, for any Church numeral \(\bar{n}\), how many nodes are there in the parse tree of \(\bar{n}\)? How many nodes are there in the parse tree of \(\bar{n}'\)?