1. Consider the following nested function all expression in Racket. Using what you know about eager evaluation, state the order in which function calls occur.

\[(a (b (c d) e (f g)) h (i) (j k l))\]

2. Here is a simple macro definition:

\[
\begin{align*}
&\textbf{(define-syntax mac} \\
&\text{(syntax-rules (allo)} \\
&\quad \text{(mac <x> allo) (* <x> <x>))})
\end{align*}
\]

Explain the output or type of error that occurs in each of the following expressions.

- \[(\text{mac 10 allo)}\]
- \[(\text{mac 20 "hi"})\]
- \[(\text{mac 20})\]
- \[(\text{mac (/ 1 0) allo)}\]
- \[(\text{mac (/ 1 0) "hi")}\]

3. Show what \[(\text{mac (+ 1 20) allo)}\] transforms into after macro expansion. Note that this is \textit{not} the same as simply evaluating the expression, which is what happens at runtime.