Exercise 1 (β-reduction)

A term \( t \) is in \( \beta \)-normal form if there is no term \( t' \) such that \( t \rightarrow^* \beta t' \). List all terms \( t \) such that:

\[
(\lambda x. (\lambda x y. x) z y) ((\lambda x. x x) (\lambda x. x x) ((\lambda x y. x) y)) \rightarrow^* t
\]

Which are normal forms?

Exercise 2 (Lists in λ-calculus)

Specify λ-terms for nil, cons, hd, tl and null, that encode lists in the λ-calculus. Show that your terms satisfy the following conditions:

\[
\begin{align*}
\text{null} & \quad \rightarrow^* \beta \quad \text{true} & \quad \text{hd} & \quad (\text{cons} \ x \ l) & \quad \rightarrow^* \beta \quad x \\
n\text{null} & \quad (\text{cons} \ x \ l) & \quad \rightarrow^* \beta \quad \text{false} & \quad \text{tl} & \quad (\text{cons} \ x \ l) & \quad \rightarrow^* \beta \quad l
\end{align*}
\]

Hint: Use pairs.
Homework 3 (Substitution Lemma)
Show that, given $x \neq y$ and $x \notin \text{FV}(u)$:

$$s[t/x][u/y] = s[u/y][t[u/y]/x]$$

Homework 4 (Trees in $\lambda$-calculus)
Encode a datatype of binary trees in lambda calculus. Specify the operations \textbf{tip} and \textbf{node} that construct trees, as well as \textbf{isTip}, \textbf{left}, \textbf{right}, and \textbf{value}. Each tip should carry a value, whereas each node should consist of two subtrees.

Show that the following holds:

- \textbf{isTip} (\textbf{tip} a) $\rightarrow^*_{\beta}$ true
- \textbf{isTip} (\textbf{node} x y) $\rightarrow^*_{\beta}$ false
- \textbf{value} (\textbf{tip} a) $\rightarrow^*_{\beta}$ a
- \textbf{left} (\textbf{node} x y) $\rightarrow^*_{\beta}$ x
- \textbf{right} (\textbf{node} x y) $\rightarrow^*_{\beta}$ y

Homework 5 (Alternative Encoding of Lists)
In this exercise, we consider an alternative encoding of lists. The list $[x, y, z]$, for instance, will now be encoded as: $\lambda c n. c x (c y (c z n))$ (speaking in terms of functional programming, each list now encodes its corresponding \textit{fold}). As in the tutorial, define the functions \textbf{nil}, \textbf{cons}, \textbf{hd}, and \textbf{null} for this encoding and show that they satisfy the same conditions. You do not need to define \textbf{tl}.

Homework 6 (Multiplication)
Define multiplication as a closed $\lambda$-term using \textbf{add} but no other helper functions and demonstrate its correctness on an example.