Exercise 1 ($\beta$-reduction)
List all terms $t$ such that:

$$\left(\lambda x.\left(\lambda xy.\left(\lambda xx\right)\left(\lambda x.\left(\lambda xy.\left(\lambda xx\right)y\right)\right)\right)z\right)
\rightarrow^* t$$

Which are normal forms?

Exercise 2 (Lists in $\lambda$-calculus)
Specify $\lambda$-terms for $\text{nil}$, $\text{cons}$, $\text{hd}$, $\text{tl}$ and $\text{null}$, that encode lists in the $\lambda$-calculus. Show that your terms satisfy the following conditions:

$$\begin{align*}
\text{null } \text{nil} & \rightarrow^* \text{true} \\
\text{null } \text{(cons } x \text{ l}) & \rightarrow^* \text{false} \\
\text{hd } \text{(cons } x \text{ l}) & \rightarrow^* x \\
\text{tl } \text{(cons } x \text{ l}) & \rightarrow^* l
\end{align*}$$

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 $\text{tip}$ and $\text{node}$ that construct trees, as well as $\text{isTip}$, $\text{left}$, $\text{right}$, and $\text{value}$. Each tip should carry a value, whereas each node should consist of two subtrees.

Show that the following holds:

$$\begin{align*}
\text{isTip } \text{(tip } a) & \rightarrow^* \text{true} \\
\text{isTip } \text{(node } x \ y) & \rightarrow^* \text{false} \\
\text{value } \text{(tip } a) & \rightarrow^* a \\
\text{left } \text{(node } x \ y) & \rightarrow^* x \\
\text{right } \text{(node } x \ y) & \rightarrow^* y
\end{align*}$$
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: \(\lambda cn.cx(\text{cy}(\text{czn}))\) (speaking in terms of functional programming, each list now encodes its corresponding fold). As in the tutorial, define the functions \(\text{nil}\), \(\text{cons}\), \(\text{hd}\), and \(\text{null}\) for this encoding and show that they satisfy the same conditions. You do not need to define \(\text{tl}\).

Homework 6 (Multiplication)

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