1. Consider the example of section 6.5.4. Redefine the calculator object so that only field update, and not proper method update, is used. The new object should have the same behavior as the original one (at least for normal users). (Exercise due to Martin Abadi.)

2. Section 6.3.1 shows that is possible to encode the untyped Lambda calculus in the Sigma calculus.
   
   (a) Show how to encode the function $\text{apply} \triangleq \lambda(f)\lambda(x)f(x)$. That is, use the translation to get $\langle\langle \text{apply} \rangle\rangle$.
   
   (b) Using the encoding from part (a) show that $\langle\langle \text{apply} \ (\lambda(x)x) \ (y) \rangle\rangle \rightarrow y$.

3. Show that the interpreter presented in Section 6.2.5 is sound with respect to the operational semantics. That is, show that if $\text{Outcome}(c) \equiv v$ and $v$ is not wrong then $\vdash c \rightarrow v$.

4. Reduction in the Sigma calculus (Section 6.2):

   (a) Find two terms $a$ and $b$ such that $a \rightarrow b$ but not $a \rightarrow\rightarrow b$.
   
   (b) Find a term $a_0$ such that $a_0 \rightarrow a_1 \rightarrow \ldots$ where $\text{length}(a_{i+1}) > \text{length}(a_i)$ for $i \geq 0$.
       Define $\text{length}(a)$ as the number of symbols in $a$.
   
   (c) Find a term that normalizes but not strongly.

5. Find example showing why $v$ and $v'$ are not necessarily the same in the statement of Theorem 6.2-4 (Completeness of weak reduction). That is, show that if $v$ and $v'$ are required to be the same, then the theorem does not hold.