1. **Prove or disprove the following statements:**

   (a) The function f : N -> N which is undefined for all even arguments and 0 for all even arguments is computable.

   Yes, the following Haskell program computes it:

   f x = if (even x) then (f x) else 0

   By Church's thesis this is then computable

   (b) If an open expression in lambda-calculus has a normal form, then this normal form is open.

   No, we can take the open expression (\x.y. x) (\x.x) z. It computes to the closed expression \x.x

   (c) The set of total functions Bool -> N is enumerable.

   Intuitively, this set is can be put in a 1-1 correspondence with the set NxN which enumerable. Each element f in the set can be seen as the pair (f(true), f(false)). An injective function from the set to N is

   g(f) = exp (2, f(true)) x exp(3, f(false))

   where exp is the exponential function.

   (d) If we fully evaluate a program in X which has a weak head normal form then the evaluation terminates.

   No, not in general. There is a simple example in the lecture notes.

2. **There are two different definitions of what it means for a set A to be enumerable. A third definition could be that the set is enumerable if it fits in a hotel, with an infinite number of rooms. Each room number is a unique natural number. That the set fits in the hotel means that (1) each element is in at least one room, and (2) each room has not more than one element (we don't want all elements to share the same room). Explain how these two requirements are satisfied in the two alternative definitions of enumerable!**

   This is explained in the lecture notes

3. **What does it mean that a function f : N -> N is computable in lambda calculus?**

   It means that there is a function g in lambda calculus such that

   g a' reduces to b' if and only if f(a) = b, where a' and b' are representations of the numbers a and b, respectively.

4. **Define a program Y in X such that the application (Y f) computes to the same value as the application (f (Y f)) for all functions f. (Don't forget to explain why!)**

We want to find a program Y such that

(Y f) = (f (Y f))

i.e. Y = \f. (f (Y f))

such equations can be solved by the program

rec Y = \f. (f (Y f))