1. We defined the Church numerals in lecture as
\[
0 := \lambda f. \lambda x. x \\
1 := \lambda f. \lambda x. f x \\
2 := \lambda f. \lambda x. f (f x)
\]
and so on, with \( n \) corresponding to applying the function \( f \) iteratively \( n \) times on \( x \). CMU Professor Emeritus and 1976 Turing Award winner Dana Scott defined, in the 1960’s, numerals in the following alternate way:
\[
0 := \lambda f. \lambda x. x \text{ (the same as Church numeral)} \\
1 := \lambda f. \lambda x. f 0 \\
2 := \lambda f. \lambda x. f 1 
\]
and so on.

(a) Write down a lambda expression that serves the role of the successor function \( \text{Succ} \) for the Scott numerals.

(b) The Scott numerals have the property that \( n \ E \ F = F \) if \( n = 0 \), and \( n \ E \ F = E m \) if \( n = m + 1 \). This has the advantage that the predecessor of a numeral can be defined readily. Can you specify a lambda expression for the predecessor function \( \text{Pred} \)?

(c) Verify that your lambda expressions above satisfy \( \text{Pred} (\text{Succ} n) = n \) for all integers \( n \geq 0 \).

(d) Give a lambda expression implementing the \( \text{isZero} \) function for Scott numerals, and argue why your expression satisfies \( \text{isZero} \ n \) is TRUE when \( n = 0 \) and FALSE when \( n > 0 \) (here TRUE and FALSE are the Boolean values defined in lecture).