**Part 1: Lazy Evaluation**

**Question 1:**

What is displayed and returned when the code below is executed?

```
(define proc
  (lambda (x)
    (display "\n")
    x))

(define foo1
  (lambda ((x lazy) y (z lazy -memo))
    (+ x y)
    (+ y z)
    (+ x z)))

(foo1 (proc 1) (proc 2) (proc 3))
```

**Question 2:**

Does anything change when the following code is executed?

```
(define foo2
  (lambda ((a lazy-memo) (b lazy-memo) (c lazy-memo))
    (foo1 a b c)))

(foo2 (proc 1) (proc 2) (proc 3))
```

**Part 2: Streams**

**Question 1:**

List the first five elements of stream st4:

```
(define st1 (cons-stream 1 st1))
(define st2 (cons-stream 1 (add-streams st1 st2)))
(define st3 (add-streams st2 st2))
(define st4 (cons-stream 0
               (add-streams st4 st3)))
```

**Question 2:**

A simplified predator/prey relationship between rabbits and foxes can be modeled by the following difference equation:

\[
\begin{align*}
\#Rabbits[n+1] &= (1 + K_{birth})\#Rabbits[n] - K_{encounter}\#Rabbits[n]\#Foxes[n] \\
\#Foxes[n+1] &= (1 - K_{death})\#Foxes[n] + K_{encounter}\#Rabbits[n]\#Foxes[n]
\end{align*}
\]

Using map2-stream, create 2 streams, Rabbits and Foxes that model this system (use variables Kb, Ke, and Kd for the constants).