Intro to Computer Science
Final Exam Review

I. Running and Writing Programs
   A. Functions that produce images
   B. Numbers and arithmetic
   C. Strings
   D. Combining functions
   E. Errors
   F. Variables
   G. **Design Recipe**
      1. Contract & Purpose Statement
      2. Function Header
         a. parameters
         b. arguments
      3. Function Body
         a. skeleton
         b. inventory
      4. Test Cases
         a. check-expect
         b. check-within
         c. check-range
   H. Animations
      5. Draw handler
      6. Tick handler

II. Definition by Choices

   A. Booleans

   B. Conditionals

   C. Animations

      a. Check-with

      b. Key handler

      c. Mouse handler

      d. Stop-when

III. Definition by Parts

   A. Structures

      a. posn

      b. color

      c. Functions using posn or color

      d. Animations using posn or color

   B. Defining Structures

      a. Functions “for free”

      b. Writing functions on user-defined structures

      c. Animations using user-defined structures

IV. Definition by Self-Reference

   A. Lists

      a. Defining lists

      b. Functions on lists

      c. Recursion

      d. Lists of structs

      e. Strings as lists

      f. Nested lists

   B. Functions that return lists

      a. Making decisions on each element

      b. Animations with lists

      c. Strings as lists

      d. Whole Numbers

Sample Programming Exercises

1. Develop structure and data definitions for a collection of zoo animals. The collection includes

   Spiders, whose relevant attributes are the number of remaining legs (we assume that spiders can lose legs in accidents) and the space (in cubic feet) they need in case of transport;

   Elephants, whose only attributes are the space (in cubic feet) they need in case of transport;

   Monkeys, whose attributes are intelligence and space (in cubic feet) needed for transportation.

   Develop the function ***fits?***. The function consumes a zoo animal and the volume of a cage. It determines whether the cage is large enough for the animal.

2. Develop an animation of an airplane that flies across the animation window from left to right. It should start at the top-left corner of the animation window and each time that it reaches the right side of the window, it should wrap around to the left side of the screen, but a few pixels closer to the ground at the bottom of the screen. There should be keys that allow the user to control the vertical movement of the airplane and the "q" key should end the animation. The animation should also end when the airplane reaches the ground at the bottom of the screen.