CSE 341, Autumn 2006, Assignment 1
Scheme Warmup
Due: Wed October 4, 10:00pm

10 points total

1. Write a function `cone-volume` that takes two real numbers representing the height and radius of the base of a cone, and that returns the volume of the cone. To head off some of the usual questions: "What's the formula for the volume of a cone?" Ans: look it up. (OK, that wasn't super-helpful, was it?) "What value of π should we use?" Ans: use any reasonable number of decimal places, say at least 4. Or better, poke around and see if π is defined as a constant in DrScheme.

2. Write a recursive function `cubes` that takes a list of integers, and returns a list of the cubes of those integers. For example, `(cubes '(1 4))` should evaluate to `(1 64)`, while `(cubes ())` should evaluate to `()`.

3. Write a function `repeat` that takes any value x and an int n, and returns a list containing n occurrences of x. For example, `(repeat 'clam 4)` should return `(clam clam clam clam)`, and `(repeat 'clam 0)` should return `()`. If n is negative, also return an empty list.

4. Write a function `repeatlist` that takes a list and an int n, and returns a new list that is n times as long, with each element repeated n times. For example, `(repeatlist '(peter paul mary) 2)` should return `(peter peter paul paul mary mary)`, and `(repeatlist () 10)` should return `()`. (Use the function `repeat` you wrote for Question 3 in defining `repeatlist`.)

5. Write a function to test whether a list of ints is in strict ascending order. For example, `(ascending '(1 2 3))` should return `#t`, while `(ascending '(2 3 1))` and `(ascending '(2 2))` should both return `#f`. You should handle the empty list, and a list of one number. (What should these return? Justify your decision in a comment in the code.)

To make it easy for the TAs to grade your homeworks, also define a function `run-all` (with no arguments) that runs test cases for all of your functions above and prints out the results in a readable way. (Hint: use the built-in functions `display` and `newline` as needed.)

Turnin:
Turn in the source listing for your functions (including `run-all`), all in one file. You should exhibit thorough tests that exercise the different cases. For example, if your function takes a list as an argument, test it with an empty list, a list with one element, and a list with several arguments. As another example, if your function takes a number, include tests that exercise the different possibilities (is something special supposed to happen with 0 or a negative number as an argument, for example)?

Assessment: Your solutions should be:

- Correct
- Tastefully commented
- In good style, including indentation and line breaks
- Written in a functional style (no side effects).
- Well tested

Your solution should be quite short — this is not a long assignment!