

Due date: Thursday, April 11, 2019, before midnight, upload in home folder of class.

Scheme interpreter instructions: For Scheme, log onto on the lab machines. Type `csi` in the terminal. It will open the chicken interpreter. (Note: The chicken interpreter uses lower case letters for all reserved words in Scheme. The book example code use uppercase for reserved words. So when you are using chicken scheme use lowercase for reserved words. For example: LAMBDA will be `lambda` in chicken interpreter.)

Please do not copy solutions from the web. Points will be deducted for copied solutions.

- a) What does the following Scheme function do for a simple list (no lists within lists allowed)? Explain. Run this code on an example input list and show the output.

```
(define (myguess lis)
  (cond
    ((null? lis) 0)
    (else (+ (car lis) (myguess (cdr lis))))
  ))
```

[4 points]

- b) What does the following Scheme function, that allows lists within lists, do? Explain. Run this code on an example input and show the output.

```
(define (myguess2 lis)
  (cond
    ((null? lis) 0)
    ((not (list? (car lis))) (+ (car lis) (myguess2 (cdr lis))))
    (else (+ (myguess2 (car lis)) (myguess2 (cdr lis))))
  ))
```

[4 points]

- c) Write a Scheme function that returns the number of threes in a given simple list of numbers.

[4 points]

- d) Repeat programming exercise c), except that you can have lists inside lists. [4 points]

- e) Write a Scheme function using `map` and `lambda` that returns the cube of each element in an example list. For instance, if the list is `(1 2 3)` the lambda function will return `(1 8 27)`.

[4 points]