

Due date: Tuesday, April 14, 2020, before midnight, upload in home folder of class.

Note: The total points here is set to 20, but the assignment will be weighted in the final scores to capture also the points of the third quiz.

a) [3 points] 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))))
  ))
```

b) [3 points] 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))))
  ))
```

c) [3 points] Write a Scheme function that returns the number of ones in a given simple list of numbers.

d) [3 points] Repeat programming exercise c), except that you can have lists inside lists.

e) [2 points] 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).

f) [3 points] How do purely functional languages differ from imperative languages? Also, it is interesting that in recent years some functional language capabilities have been becoming available in imperative languages. Give a brief example of a functional capability in an imperative language.

g) [3 points] Compare the list processing capabilities of Scheme to ML or Haskell. In what ways do they differ and in what ways are they similar?