

CS 101 Computer Science I (Spring 2001)
Lab 4: February 14-16, 2001

1. Write a deeply recursive Scheme procedure called "**sum***" that takes either a number or a nested list of numbers as input. The procedure **sum*** returns the sum of all the numbers appearing anywhere in the input expression.

```
(sum* '())           ==> 0
(sum* 7)            ==> 7
(sum* '(2 3 0 5))  ==> 10
(sum* '(-3 (1 (8)) 7)) ==> 13
```

2. Write a deeply recursive Scheme procedure called "**count-odd***" that takes either an integer or a nested list of integers as input. The procedure **count-odd*** returns the number of odd integers appearing appearing anywhere in the input expression.

```
(count-odd* '())           ==> 0
(count-odd* 7)            ==> 1
(count-odd* 8)            ==> 0
(count-odd* '(2 3 0 5))  ==> 2
(count-odd* '(-3 (1 (8)) 7)) ==> 3
```

3. Write a deeply recursive Scheme predicate procedure called "**all-even*?**" that takes either an integer or a nested list of integers as input. The procedure **all-even*?** returns **#t** if all of the integers appearing in the input expression are even. Otherwise it returns **#f**.

```
(all-even*? '())           ==> #t
(all-even*? 7)            ==> #f
(all-even*? 8)            ==> #t
(all-even*? '(2 3 0 5))  ==> #f
(all-even*? '(2 4 0 6))  ==> #t
(all-even*? '(-3 (1 (8)) 7)) ==> #f
(all-even*? '(-4 (0 (8)) 6)) ==> #t
```