CS2500 Exam 2

Name: ____________________________________

Student Id (last 4 digits): ____________________________________

- Write down the answers in the space provided.
- You may use the usual primitives and expression forms, including those suggested in hints; for everything else, define it.
- The phrase "design this function/program" means that you should apply the design recipe. You are *not* required to provide a template unless the problem specifically asks for one. Be prepared, however, to struggle with the development of function bodies if you choose to skip the template step.

Problem	Points	out of
1		8
2		10
3		10
4		10
Extra		5
Total		38

Good luck!

Problem 1. Suppose we have the following list:

```
 (define x ‘((a b) (1 2) (3 4) (5 6)))
```

What does each of the following expressions produce?

```
a. (map rest x)
```

```
b. (filter cons? x)
```

```
c. (andmap symbol? (map first x))
```

```
d. (foldr + 0 (apply append (rest x)))
```

Problem 2.

Study the definition for `foo` below, and give it a general contract.

```
(define (foo g f par xs)
  (cond
    [(empty? xs) empty]
    [(g (first xs))
        (cons (f (first xs) par)
              (foo g f par (rest xs)))]
    [else (foo g f par (rest xs))]))
```

Problem 3. Due to an unfortunately timed bug, some of DrRacket's built in loop functions have become unreliable the night before your assignment is due! The (partial) good news is that the `foldr` function still works.

It would be really handy to use `andmap` in your assignment. Since you don't have time to wait for a DrRacket patch to be developed that will fix the issue, you will have to define it yourself. Fortunately, you are a good enough programmer to realize that you can write `andmap` rather simply using `foldr`. You may use `lambda` or `local`, if needed.

Design `andmap` using `foldr`.

Problem 4. Consider the following data definition:

```
(define-struct student (name lab awake? quizzes))
;; A Student is a (make-student String Symbol
;;                           Boolean [Listof Number])
;; where:  lab      - one of ‘mon or ‘wed
;;         awake?   - true if the student asks
;;                    and answers questions in class
;;        quizzes   - the list of grades assigned
;;                    for the class quizzes
```

a) Design a function, `sleepy-students`, that consumes a `[Listof Student]`, and returns the `[Listof Student]` that are not awake. Use a loop function (map, foldr, filter, etc).

b) Design a function, `list-names`, that takes a [`Listof Student]` and returns a list of their names. Use a loop function (map, foldr, filter, etc).

Problem 5. Consider the following data definitions:

```
;; a Record is a
;; (make-record String [Listof Number])
(define-struct record (name log))
;; where log is a [Listof Number] representing a
;; list of exam grades
```

Here are two examples:

```
(define r1 (make-record "Mary" '(80 90)))
(define r2 (make-record "Bob" '(82 85)))

;; an Exam is a (make-exam String Number)
(define-struct exam (name grade))
```

Here are two examples:

```
(define e1 (make-exam "Mary" 92))
(define e2 (make-exam "Bob" 89))
```

Design a function `add-grade` that consumes a `[Listof Record]` and a `[Listof Exam]` and adds the exam grade to each student's `[Listof Grade]`
ASSUMPTION: the two lists are of equal length
ASSUMPTION: the student names in the two lists are in the same order

Here is a test to help you understand what the function should do:

```
(check-expect
    (add-grade (list r1 r2) (list e1 e2))
       (list (make-record "Mary" '(92 80 90))
             (make-record "Bob" '(89 82 85))))
```