

CSC 324H1 S March Midterm 2 2014
Duration — 50 minutes

Student Number:

Family Name: Given Name:

*Do **not** turn this page until you have received the signal to start.*
(Please fill out the identification section above,
and read the instructions below.) *Good Luck!*

1: / 6

2: / 7

This quiz consists of 4 questions on 4 pages (including this one). *When you receive the signal to start, please make sure that your copy is complete.*
If you do any rough work, indicate clearly what you want marked.

3: / 13

4: / 10

TOTAL: / 36

Question 1. [6 MARKS]

Recall:

```
(check-expect (max 3) 3)
(check-expect (max 3 200 40) 200)
(check-expect (append) '())
(check-expect (append '(a 1)) '(a 1))
(check-expect (append '(2 b c) '(a 1) '(e)) '(2 b c a 1 e))
```

For each of the following expressions, write its result value as simply as possible:

```
(apply max '(3 2 4))

(map max '(3 2 4))

(apply append '((a) (b c d) (e f)))

(map append '((a) (b c d) (e f)))

(apply list '(3 2 4))

(map list '(3 2 4))
```

Question 2. [7 MARKS]

Write `depth` that takes a value and returns how deeply nested its list structure is. Assume the value is not the empty list, and the empty list does not appear in any part of the value. Use higher-order functions where appropriate. Do not do first-rest recursion manually.

```
(check-expect (depth 'a) 0)
(check-expect (depth '(a b c)) 1)
(check-expect (depth '((a b) c)) 2)
(check-expect (depth '((a b) ((c)))) 3)
```

Question 3. [13 MARKS]**Part (a)** [6 MARKS]

Write `sum` that takes two unary functions and returns their sum, i.e. returns a unary function that takes a value and returns the sum of the two numbers that the functions produce for that value.

```
(define f (sum sqr (lambda (x) (* 2 x))))  
(check-expect (f 3) 15)
```

Part (b) [7 MARKS]

Consider the following Run-Time Data Language for calculus functions:

```
sin  
cos  
exp  
(f + g) ; where f and g are in the language  
(f * g) ; where f and g are in the language
```

Write a function `cf-to-fn` that takes an expression in the data language and returns an actual unary function that calculates it. Use pattern-matching in your implementation.

```
(define g (cf-to-fn '((cos + cos) * (sin + cos))))  
(check-expect (g 0) 2)
```

Question 4. [10 MARKS]**Part (a)** [4 MARKS]

Recall:

```
(define (fix-1st f 2nd) (lambda (2nd) (f 1st 2nd)))  
(define (fix-2nd f 1st) (lambda (1st) (f 1st 2nd)))
```

Write a binary function `x` that takes a value `e` and list `L`. It returns a list of two-elements lists, where each two-element list has `e` as its first element and an element of `L` as its second element.

Use higher-order functions and the above two given functions when appropriate.

Do not do first-rest recursion manually.

```
(check-expect (x 1 '(2 3 4)) '((1 2) (1 3) (1 4)))
```

Part (b) [6 MARKS]

Write a binary function `X` taking two lists `L1` and `L2`, and returning their cartesian product: a list of every possible two-element list containing an element of `L1` as first element and an element of `L2` as second element. Use higher-order functions and the given functions when appropriate.

Do not do first-rest recursion manually.

Total Marks = 36