

PLEASE HAND IN

UNIVERSITY OF TORONTO
FACULTY OF ARTS AND SCIENCE

TERM TEST #2

CSC 104H

DURATION — 50 MINUTES

PLEASE HAND IN

LAST/FAMILY NAME: _____

FIRST/GIVEN NAME: _____

Do NOT turn this page until you have received the signal to start.
(In the meantime, please fill out the identification section above, and read the instructions below.)

This test consists of 4 questions on 6 pages (including this one).
When you receive the signal to start, please make sure that your copy of the test is complete.
Please answer questions in the space provided.
You will earn 20% for any question you leave blank or write "I cannot answer this question," on.

Good Luck!

QUESTION 1. [9 MARKS]

Assume the following has been run:

```
(define image (circle 10 "solid" "black"))
(define pixels (image->color-list image))
(define (i a-color)
  (make-color (- 255 (color-red a-color))
              (- 255 (color-green a-color))
              (- 255 (color-blue a-color))))
(define (nth-pixel n)
  (list-ref pixels n))
```

Show the intermediate steps and result value for the following expression.

When pixels is used in an intermediate step don't write out the list of pixels (just use the name pixels).

```
(nth-pixel 0)
```

Show the intermediate steps and result value for the following expression.

You don't need to show the steps for (nth-pixel 0) again.

```
(i (nth-pixel 0))
```

Show the result values for the following expressions:

```
(color-list->bitmap (map i pixels) (image-width image) (image-height image))
```

```
(color-list->bitmap (map nth-pixel (range 0 (length pixels) 3))
                  (/ (image-width image) 3) (image-height image))
```

QUESTION 2. [6 MARKS]

PART (A) [3 MARKS]

Convert 44 to its binary representation, BRIEFLY showing the steps of your procedure.

PART (B) [3 MARKS]

Convert the binary representation 1010011 to its decimal representation, BRIEFLY showing the steps of your procedure.


QUESTION 3. [15 MARKS]



PART (A) [4 MARKS]


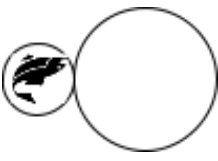
Define function `c` according to the check-expect and description, and fill in its contract.


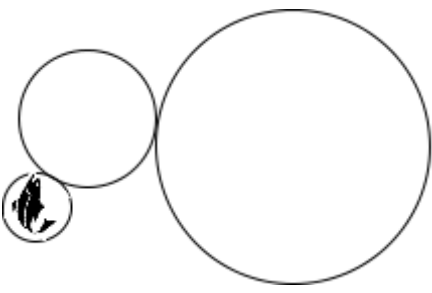
```
(check-expect (c 10) (circle 5 "outline" "black"))  
; c :          ->  
; Black outline circle with the given DIAMETER.
```

PART (B) [6 MARKS]

Assume fish has been defined: (define fish ) Here are three check-expects for a function bubble:

(check-expect (bubble  0) )

(check-expect (bubble  1) )

(check-expect (bubble  2) )

Complete the following three check-expects WITHOUT drawing any images manually by hand.


Use c from Part (A) to help you.


Use fish inside your first check-expect.


Use (bubble fish 0) inside your second check-expect.

Use (bubble fish 1) inside your third check-expect.

You may assume the width and height of the fish image are equal.

(check-expect (bubble  0))

(check-expect (bubble  1))

(check-expect (bubble  2))

PART (B) [4 MARKS]

Reminder:

```
(define (q a-list)
  (cond [(= (length a-list) 1) (first a-list)]
        [else (above (beside (q (rest a-list))
                              (q (rest a-list)))
                      (first a-list))]))
```

Show the result of the following expression, briefly showing your steps [include at least two illustrative intermediate steps]:

```
(q (list (circle 5 "outline" "black")
        (triangle 10 "outline" "black")
        (square 10 "outline" "black")))
```

1: _____/ 9

2: _____/ 6

3: _____/ 15

4: _____/ 8

TOTAL: _____/ 38