Do not turn this page until you have received the signal to start.
In the meantime, please read the instructions below carefully.

This Final Examination paper consists of 8 questions on 13 pages (including this one), printed on both sides of the paper. When you receive the signal to start, please make sure that your copy of the paper is complete and fill in your name and student number above.

Answer each question directly on this exam paper, in the space provided or on the facing blank page. If you need more space for one of your solutions you may also use a “blank” page at the end of the paper (in which case make sure to mention that where the question is asked).

You will earn 20% for any question you leave blank or you write “I cannot answer this question”.

Marking Guide

# 1: _____/ 9
# 2: _____/10
# 3: _____/ 9
# 4: _____/ 8
# 5: _____/10
# 6: _____/ 4
# 7: _____/ 7
# 8: _____/12

TOTAL: _____/69
Question 1. [9 marks]

Part (a) [2 marks]
Convert the binary representation 10001111 to its decimal representation, briefly showing your steps.

Part (b) [4 marks]
Multiply the numbers whose binary representations are 1011 and 1101, using the multiplication algorithm in binary notation, showing your work.

Part (c) [3 marks]
The number 3000 has the binary representation 101110111000.
Use that to show the the binary representations of 1500, 6000, and 6002.
**Question 2.** [10 marks]

Consider the following program:

```
(require picturing-programs)

(define T (triangle 10 "outline" "black"))
(define TT (triangle 10 "solid" "black"))

(define-struct status (a d))

(define (draw s)
  (rotate (* 120 (status-a s))
    (above T (beside T TT))))

(define (update s)
  (cond [(= (status-a s) 0) (make-status 1 1)]
        [(= (status-a s) 2) (make-status 1 -1)]
        [else (make-status (+ (status-a s) (status-d s)) (status-d s))])))
```

For each of the following expressions, show the result value.
For the **big-bang** expression, show the first four frames of the animation.

T

TT

(draw (make-status 0 1))

(update (make-status 0 1))

(update (make-status 1 1))

(big-bang (make-status 0 1)
  (on-draw draw)
  (on-tick update))
Question 3. [9 MARKS]

Part (a) [4 MARKS]
Show the result values of each of the following expressions [except the require and definitions].
Recall: the color struct has its components in the order: RED GREEN BLUE ALPHA.
The maximum value a component can have is 255.

(require picturing-programs)
(define S (square 4 "solid" "blue"))

(length (image->color-list S))

(first (image->color-list S))

(color-blue (first (image->color-list S)))

(color-red (first (image->color-list S)))

Part (b) [5 MARKS]
Draw what is produced by the last expression in the following program. Briefly explain your reasoning.
(require picturing-programs)
(define an-image
  \begin{center}
  \includegraphics{image.png}
  \end{center}
)

(define pixels (image->color-list an-image))

; Recall: (take L n) produces the sublist of L containing the first n elements.

(color-list->bitmap
  (reverse
    (take pixels
      (/ (length pixels) 2)))
  ; These are in the usual order for color-list->bitmap .
  (image-width an-image) (/ (image-height an-image) 2))
)
Question 4. [8 marks]

Part (a) [3 marks]
Define the function \( c \) according to the check-expect and description. Also, fill in its contract.

(require picturing-programs)

(check-expect (c 10) (circle 10 "outline" "black"))

; \( c : \cdot \rightarrow \cdot \)
; Black outline circle of the given radius.

Part (b) [3 marks]
Recall the function range. Here's an example of it:
(check-expect (range 12 34 5) (list 12 17 22 27 32))
Use \( c \) and range to define circles:
; The first circle has radius 10 and the last one has radius 50.
(check-expect circles

   (list ⫋ ⫋ ⫋ ⫊)

Part (c) [2 marks]
Give an expression using circles, to produce the image:
Question 5. [10 marks]
Consider the function $s$:

\[
\text{(require picturing-programs)}
\]
\[
; s : \text{number} \rightarrow \text{image}
\]
\[
(\text{define (s d)}
\]
\[
(\text{cond [(=} d 0) (\text{square 10 "outline" "black"}])}
\]
\[
[\text{else (overlay}}
\]
\[
(\text{above (s (- d 1))}
\]
\[
(s (- d 1)))
\]
\[
(\text{square (* 2 (image-width (s (- d 1))))}
\]
\[
"\text{outline" "black"})])]
\]

Part (a) [4 marks]
Draw the values of $(s \ 0)$ and $(s \ 1)$.

Part (b) [6 marks]
Draw the result value of $(s \ 3)$, briefly showing your steps.
Question 6.  [4 marks]
Consider the following definitions:

(define (over-26? im)
  (> (image-width im) 26))

(define (over-4? s)
  (> (string-length s) 4))

(define L1 (list (square 10 "solid" "black")
  (circle 30 "outline" "black")
  (rectangle 25 100 "solid" "green")))

(define L2 (list "birds" "fly" "fish" "swim" "and" "students" "think"))

Show the result value of each of the following expressions:

(length (filter over-26? L1))

(filter over-4? L2)

Question 7.  [7 marks]
Show the result values of each of the following expressions [except definitions]:

(define lol-A (list "a" 
  (list "b" (list "c" "d") "e")
  (list "f")))

(length lol-A)

(second lol-A)

(rest lol-A)

(reverse lol-A)

(define lol-B (list (list "a" "b")
  (list "c" (list "d" "e") "f")))

(apply append lol-B)
Question 8. [12 marks]
Part (a) [6 marks]
Consider the function \( r \):

\[
\text{(define (r lol)}
\begin{align*}
\text{  (cond [((list? lol) (reverse (map r lol))])} \\
\text{  [else lol])])}
\end{align*}
\text{)}
\]

Show the result values of each of the following expressions. For the third one, show how.

\((r 1)\)

\((r \text{(list 2 3)})\)

\((r \text{(list 1 (list 2 3))})\); Show some intermediate steps for this one.

\((r \text{(list (list 1 (list 2 3))))})\)
Part (b)  [6 marks]
Consider the function \( f \) :

\[
\text{(define (f lol)}
\begin{align*}
&\quad \text{(cond \[(\text{list? lol)} \ (\text{reverse \ (apply \ append \ (map \ f \ lol))})
\\quad \text{[else \ (list \ lol)])})
\end{align*}
\)
\]

Show the result values of each of the following expressions. For the third one, show how.

(f 1)

(f (list 1 2))

(f (list (list 1 2) 3 4)) ; Show some intermediate steps for this one.
Use the space on this “blank” page for scratch work, or for any answer that did not fit elsewhere.

Clearly label each such answer with the appropriate question and part number.
Use the space on this “blank” page for scratch work, or for any answer that did not fit elsewhere.

**Clearly label each such answer with the appropriate question and part number.**
Please write nothing on this page.