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 5 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. [4 marks]

Some people claim: software is programmed by humans, so software will always act in ways understandable by humans.

Mention something covered in the course (for example a built-in racket function, an algorithm, a problem to solve, etc) that could make someone doubt that claim. Briefly explain why.

QUESTION 2. [6 marks]

Complete the table below. In each row, assume a, b, and c have the given values, and then evaluate the expression at the end of the row — it will be either true or false.

<table>
<thead>
<tr>
<th>a</th>
<th>b</th>
<th>c</th>
<th>(and (not a) (or b c))</th>
</tr>
</thead>
<tbody>
<tr>
<td>true</td>
<td>true</td>
<td>true</td>
<td>true</td>
</tr>
<tr>
<td>true</td>
<td>true</td>
<td>false</td>
<td>false</td>
</tr>
<tr>
<td>true</td>
<td>false</td>
<td>false</td>
<td>false</td>
</tr>
<tr>
<td>false</td>
<td>true</td>
<td>true</td>
<td>true</td>
</tr>
<tr>
<td>false</td>
<td>false</td>
<td>true</td>
<td>true</td>
</tr>
<tr>
<td>false</td>
<td>false</td>
<td>false</td>
<td>false</td>
</tr>
</tbody>
</table>
QUESTION 3. [10 marks]

Assume the expressions below have been typed into the Definitions pane of DrRacket. Below each one [except (require picturing-programs)], show the value that the expression produces when the “Run” button is clicked. You can show your steps, which can earn you part marks if your final result is incorrect. The colour of any images you draw is unimportant: use whatever colour pen or pencil you have.

(require picturing-programs)

(rotate (* (string-length "hello") (string-length "bye-bye")))
   (above (circle 10 "solid" "black") (circle 20 "outline" "black")))

(image-height
   (beside (scale 3 (square 4 "solid" "red")) (square (+ 5 6) "outline" "blue")))

(- 20 (apply + (map sqr (list 1 2 3))))

(first (reverse (list "hello" (string-append "and" "bye"))))
QUESTION 4. [10 marks]

Complete the functions increasing-faster? and tall? by:

(i) Writing another check-expect expression.

(ii) Filling in the contract: the datatype of each input, followed by an arrow "->", followed by the datatype of the output. We already put in a comment line for the contract, with the "->", so you just need to write down the datatypes in the space provided.

(iii) Writing the body of the function.

PART (A) [5 marks]

; increasing-faster? : ->
;
; Is the change from y to z more than the change from x to y?
(define (increasing-faster? x y z) ; Write the body of the function below here.

)
; For example, the change from 12 to 17 is 5, but the change from 10 to 12 is just 2,
; so the change from 12 to 17 is more than the change from 10 to 12:
(check-expect (increasing-faster? 10 12 17) true)
; Write another check-expect for increasing-faster? :

PART (B) [5 marks]

; tall? : ->
;
; Is an-image taller than it is wide?
(define (tall? an-image) ; Write the body of the function below here.

)

(check-expect (tall? ♥)
    (> (image-height ♥) (image-width ♥)))
; Write another check-expect for tall? :
QUESTION 5. [10 marks]

Assume these two expressions have been typed into the Definitions pane of DrRacket:
(require picturing-programs)
(define a-cat 😻)

The following leads you to writing a new function eary.
Do not manually draw any cat faces by hand.

PART (A) [2 marks] Write an expression that produces 😻:

PART (B) [2 marks]
Write an expression that produces 😻 😻:

PART (C) [2 marks]
Write an expression that produces 😻:

PART (D) [1 mark] Complete the second check-expect for function eary:
(check-expect (eary 😻) 😻)
(check-expect (eary 😻))

PART (E) [3 marks] Write the header and body to define the eary function:
(define
# 1: _____/ 4
# 2: _____/ 6
# 3: _____/ 10
# 4: _____/ 10
# 5: _____/ 10

TOTAL: _____/ 40