CS 135 Winter 2018
Midterm Review Session
Announcements

- Assignment 05 is due tomorrow!
- The midterm is on **Monday, February 26**, from **7:00 to 8:50 PM**, and covers material up to and including Module 06, Slide 50.
- Office Hours have been rescheduled for the next few weeks: check Piazza/the course website.
Structures

Consider the following structure and data definition:

(define-struct burger (patty veg-count bun-type cheese?))

;; A Burger is a (make-burger Sym Nat Sym Bool)
;; requires: patty is one of ’veggie, chicken, or ’beef
;; veg-count <= 10
;; bun-type is one of ’white or ’whole-wheat

Write a template for a Burger and a (listof Burger).
Structures

The following are considered healthy characteristics of a Burger:

- The patty is either ’veggie or ’chicken.
- The veg-count of a Burger is at least 5.
- The bun-type is ’whole-wheat.
- The Burger does not have cheese on it (cheese? is false).

Write a function healthy?, that consumes a Burger, and produces true if that Burger has at least two of the four healthy characteristics mentioned above, and false otherwise. Avoid enumerating every single possibility that would make a burger healthy.
Lists of Structures

Write a function health-score that consumes a list of Burgers, and produces the number of burgers from the consumed list that are considered healthy, according to the characteristics above. However, for every burger in the consumed list that has zero (0) healthy characteristics, the number produced is subtracted by one (1). For example,

(health-score (list (make-burger 'veggie 4 'whole-wheat false) (make-burger 'beef 0 'white true))) would produce 0. You may use the following function:

(define (unhealthy? aburger)
  (and (symbol=? 'beef (burger-patty aburger))
       (< (burger-veg-count aburger) healthy-veg-cutoff)
       (symbol=? 'white (burger-bun-type aburger))
       (burger-cheese? aburger)))
Lists of Structures

Write a function `add-veggie-count`, that consumes a natural number, a symbol that is one of ’veggie, ’chicken, ’beef, or ’all, and a list of Burgers. The function produces a list of Burgers, where each burger is modified according to the following rules:

- If the consumed symbol is ’all, then the veg-count field of every Burger in the consumed list will be incremented by the consumed natural number. Recall that a Burger cannot have a veg-count greater than 10, if incrementing the veg-count field were to exceed this threshold.

- If the consumed symbol is ’veggie, ’chicken, or ’beef, only the Burgers in the consumed list that have the same patty field as the consumed symbol should have their veg-count field incremented (in the same way as described above).

- All other Burgers in the consumed list are unaltered.

The order of the burgers in the consumed and produced lists should be the same.
Strings and Lists of Characters

Zainab is very picky about the characters in strings. As she searches through a string one character at a time, starting from the first character of the string, she modifies that string according to the following rules:

- Zainab has no issues with the empty string, and will not modify it in any way.
- Zainab loves seeing the letter Z appear in any string. Upon encountering a lower-case or upper-case Z character, the single occurrence of a Z will be replaced with two consecutive Z’s, both with the same case as the original Z.
- Zainab does not like the letter E, as she has a hard time writing them out. As soon as Zainab sees a single occurrence of a lower-case or upper-case e, it is removed from the string. Zainab will stop searching through the rest of the string at that point, and simply produce the rest of the characters in the string (even if they should have been removed or altered according to her rules).
Zainab insists that all other upper-case characters are replaced with their corresponding lower-case characters. For example, the upper-case letter S would be replaced with the lower-case letter s in the string.

Finally, all other characters in the string will remain unchanged according to Zainab.

Write a function Zainab-convert, that consumes a string, and produces the result of modifying that string according to Zainab’s rules above. Note that it is appropriate to have a capital letter in the function name, as Zainab is a proper noun. You may find the functions char-ci=?, char-upper-case?, and char-downcase helpful.
Clicker Question: List Abbreviations

(define question3 (cons "blue"

  (cons (cons "green" (cons 57 (cons 38 empty)))
    (cons 'red
      (cons (cons 'question (cons 'three empty))
        (cons empty empty))))))

Which of the following is a correct representation of question3?

A  (list "blue" (list (list "green" 57 38) 'red) (list (list 'question 'three)) (list))
B  (list "blue" (list (list "green" 57 38)) 'red (list (list 'question 'three)))
C  (list "blue" (list "green" 57 38) 'red (list 'question 'three) (list))
D  (list "blue" (list "green" 57 38) 'red (list 'question 'three))
E  (list "blue" (list (list "green" 57 38) 'red) (list 'question 'three) (list))
Clicker Question: Lists

(define question3 (cons "blue"
    (cons (cons "green" (cons 57 (cons 38 empty)))
        (cons 'red
            (cons (cons 'question (cons 'three empty))
                (cons empty empty))))))

Which of the following is a correct representation of question3?

A  '("blue" (("green" 57) 38) red (question three) ()

B  '("blue" '("green" 57 38) red '(question three) ()

C  '("blue" (("green" 57) 38) 'red ('question 'three))

D  '("blue" ("green" 57 38) red (question three))

E  '("blue" ("green" 57 38) red (question three) ()

CS 135 Winter 2018 Midterm Review Session 10
Clicker Question: Lists

(define question3 (cons "blue"

    (cons (cons "green" (cons 57 (cons 38 empty)))

    (cons 'red

    (cons (cons 'question (cons 'three empty))

    (cons empty empty)))))

What is the value of (rest (first (rest (second question2)))))?

A  This produces an error.
B  57
C  38
D  empty
E  (cons 57 (cons 38 empty))
Clicker Question: Lists

(define question3 (cons "blue"

    (cons (cons "green" (cons 57 (cons 38 empty)))

    (cons 'red

        (cons (cons 'question (cons 'three empty))

        (cons empty empty))))))

What is the value of (length (cons (second question3) (fourth question3)))?

A  This produces an error.
B  2
C  3
D  4
E  5
Recursing on Natural Numbers

Write a function \texttt{sum-odds} that consumes two integers \texttt{start} and \texttt{end}, where \texttt{start} \leq \texttt{end}, and produces the sum of all the odd integers in the range from \texttt{start} to \texttt{end} inclusive. For example, \((\texttt{sum-odds } \neg 4 \ 7)\) produces 12.
Natural Numbers

Consider the following data definition:

;; A MysteryNat is one of
;; * 1
;; * 3
;; * (+ MysteryNat 4)

Write a **template mysterynat-template** that consumes a MysteryNat.
Natural Numbers

Write a function `base-case-3?` that consumes a `MysteryNat` and produces `true` if the consumed `MysteryNat` was recursively built up from the base case of 3 according to the data definition above, and false otherwise. The function should also produce `true` if the consumed `MysteryNat` is 3 itself. You may **not** use any form of multiplication or division. Your function should use pure structural recursion and follow from the template in part (a).
Stepping  Suppose that the following definitions have been fully processed in Beginning Student with List Abbreviations:

(define-struct review (time date location))
(define start-time 1630)
(define end-time (+ start-time 190))
(define cs135 (make-review end-time "Feb 14" 'MC))
(define cs135-lst (list start-time "Feb 14" 'MC))
(define (convert val)
  (cond [(cons? val) (make-review (first val) (rest val) (empty? cs135-lst))]
        [(review? val) (list (review-location val) (review-date cs135) (third val))]
        [else val]])
(define (stepping a b c)
  (cond [(and (not (review? (make-review a b c))) c) "correct"]
        [(or (> a end-time) (> b start-time)) (convert 'cs135)]
        [else "incorrect"]))

Step through the following: (stepping end-time 1730 false)