;;
;; Constants
;;

(define GINNY)

(define ROMILDA)

(define VINCENT)

(define PANSY)

(define DRACO)

(define CHO)
(define DOBBY)

(define ROMILDA-MILL
  (make-gossip ROMILDA '(GINNY-MILL)))

(define VINCENT-MILL
  (make-gossip VINCENT '(GINNY-MILL)))

(define DRACO-MILL
  (make-gossip DRACO-MILL '(GINNY-MILL)))
(make-gossip DRACO ROMILDA-MILL VINCENT-MILL))

(define CHO-MILL
 (make-gossip CHO '() '()))

(define PANSY-MILL
 (make-gossip PANSY CHO-MILL DRACO-MILL))

;; Functions

;; RumorMill Image -> Boolean
;; Determine whether `who` is informed about a rumor that starts in `rm`.
(define (informed? rm who)
 (cond [[(empty? rm) #false]
            [(gossip? rm)
                (or (image=? who (gossip-who rm))
                    (informed? (gossip-next1 rm) who)
                    (informed? (gossip-next2 rm) who))]))

;; Alternative version:
(define (informed? rm who)
 (and (not (empty? rm))
      (or (image=? who (gossip-who rm))
           (informed? (gossip-next1 rm) who)
           (informed? (gossip-next2 rm) who))))

(check-expect (informed? '() GINNY) #false)
(check-expect (informed? GINNY-MILL PANSY) #false)
(check-expect (informed? GINNY-MILL GINNY) #true)
(check-expect (informed? PANSY-MILL GINNY) #true)
(check-expect (informed? PANSY-MILL DOBBY) #false)
;; No one tells Dobby anything.

;; RumorMill -> NonNegativeNumber
;; Determine the maximum number of days for a rumor to propagate at a rate of
;; 1 day to pass it on
(define (rumor-delay rm)
 (cond [[(empty? rm) 0]
            [(gossip? rm)
                (+ 1 (max (rumor-delay (gossip-next1 rm))
                              (rumor-delay (gossip-next2 rm))))]

(check-expect (rumor-delay '()) 0)
(check-expect (rumor-delay CHO-MILL) 1)
(check-expect (rumor-delay ROMILDA-MILL) 2)
(check-expect (rumor-delay DRACO-MILL) 3)
(check-expect (rumor-delay PANSY-MILL) 4)

;; RumorMill Image Image -> RumorMill
;; Adds newper to rm, being told by oldper
(define (add-gossip rm newper oldper)
  (cond [(empty? rm)
        ;; Base case 1: RumorMill is empty
        '()]
        [(image=? oldper (gossip-who rm))
        ;; Base case 2: Found the oldper; add newper here!
        (add-per rm newper)]
       [else
        ;; Recursive case: Keep looking for oldper
        (make-gossip (gossip-who rm)
                      (add-gossip (gossip-next1 rm) newper oldper)
                      (add-gossip (gossip-next2 rm) newper oldper))])

;; RumorMill Image -> RumorMill
;; Add newper to rm
(define (add-per rm newper)
  (cond [(empty? (gossip-next1 rm))
         (make-gossip (gossip-who rm)
                      (make-gossip newper '() '())
                      (gossip-next2 rm))]
       [else
        (make-gossip (gossip-who rm)
                     (gossip-next1 rm)
                     (make-gossip newper '() '()))])

(check-expect (add-gossip '() DOBBY GINNY) '())
(check-expect (add-gossip GINNY-MILL DOBBY GINNY)
              (make-gossip GINNY
                          (make-gossip DOBBY '() '()))
              '())
(check-expect (add-gossip (make-gossip PANSY
                                   GINNY-MILL
                                   '())
                                   DOBBY
                                   PANSY)
              (make-gossip PANSY
                           GINNY-MILL
                           '())))