Problem (3 / 6 points): It was a dark and mysterious recursion…

Consider the recursive procedure gather that takes a sentence of at least two single-character words (i.e., letters such as 'a', 'b', etc.):

;;; sent-of-ltrs is a sentence of at least 2 words that are single letters
(define (gather sent-of-ltrs)
  (cond ((empty? sent-of-ltrs) '())
        ((empty? (bf sent-of-ltrs))
         (se (first sent-of-ltrs)))
        ((equal? (first (first sent-of-ltrs))
                 (first (bf sent-of-ltrs)))
         (gather (se (word (first sent-of-ltrs)
                          (first (bf sent-of-ltrs))))
                 (bf (bf sent-of-ltrs))))
        (else
         (se (first sent-of-ltrs)
              (gather (bf sent-of-ltrs))))))

Part A (3 points). What will (gather '(a b b c d d)) return?

Part B (6 points). Write gather-hof, which behaves the same as gather but uses no explicit recursion.