;; Homeworl 1.27

(define flatten

(lambda (lst)

(if (null? lst)

'()

(if (not(list? (car lst)))

(cons (car lst) (flatten (cdr lst)))

(append (flatten (car lst)) (flatten(cdr lst)))))))

;; (flatten '(((d)) e))

;;Home work 1.28

(define merge

(lambda (loi1 loi2)

(cond

((null? loi1) loi2)

((null? loi2) loi1)

((< (car loi1) (car loi2)) (cons (car loi1) (merge (cdr loi1) loi2)))

(else (cons (car loi2) (merge (cdr loi2) loi1))))))

;;(merge '(1 4) '(1 2 8))

;;(merge '(35 62 81 90 91) '(3 83 85 90))

;;Homework 1.29

(define spilt-head

(lambda (lst n1 n2)

(if (= n2 n1)

'()

(cons (car lst) (spilt-head (cdr lst) n1 (- n2 1))))))

(define spilt-tail

(lambda (lst n n1 n2)

(if (< n n1)

(spilt-tail (cdr lst) (+ n 1) n1 n2)

(if (<= n n2)

(cons (car lst) (spilt-tail (cdr lst) (+ n 1) n1 n2))

'()))))

(spilt-tail '(1 2 3 4 5 6) 0 3 5)

(spilt-head '(1 2 3 4 5) 0 3)