(defun masprogramas(lista)

(if(consp lista)

(cond

((> (suma1(car lista)) (suma1(cadr lista))) (pprint "TARDE MAYOR"))

((< (suma1(car lista)) (suma1(cadr lista))) (pprint "NOCHE MAYOR"))

((= (suma1(car lista)) (suma1(cadr lista))) (pprint "IGUALES"))

))

)

(defun suma1(a)

(cond

((endp a) 0)

(t(+ 1 (suma1 (cdr a))))))

(defun mayor(b)

(if(consp b)

(cond

((> (rating(car b))(rating(cadr programas)))(pprint "TARDE MAYOR"))

((< (rating(car b))(rating(cadr programas)))(pprint "NOCHE MAYOR"))

((= (rating(car b))(rating(cadr programas)))(pprint "IGUALES"))

)))

(defun rating (a)

(cond

((endp a) nil)

((endp (Cdr a))(car a))

(t (max (Car a) (max(rating (cdr a)))))))

(mapcar(lambda (x) (COND

((and(listp x)(numberp (car x))(evenp (car x))) 'PAR)

((and(listp x)(numberp (car x))(oddp (car x))) 'IMPAR)

(t 'nil)))lista)

(defun obtener-menor-tiempo-recursivo (lista)

(cond

((null lista) nil)

((null (cdr lista)) (car (car lista)))

(t (min (car (car lista)) (obtener-menor-tiempo-recursivo (cdr lista))))))