Práctica 1b
Ejercicios de programación en Common Lisp

Nombre: _____

1.- Escribe una función llamada MERGE-L que reciba como argumentos 2 listas de igual longitud. La función sumará los elementos correspondientes de cada lista y regresará el producto de los numeros resultantes. Por ejemplo,

```
(MERGE-L '(1 2 3) '(2 3 4))
```

deberá regresar 105, ya que (1+2)*(2+3)*(3+4)=3*5*7=105.

2.- Considera la siguiente definición para la función CIRCULA:

```
(defun circula (lst)
  (append (rest lst)
          (list (first lst))))
```

Esta función recibe una lista como argumento y construye una lista nueva tomando el primer elemento del argumento para convertirlo en el último de la nueva lista. Por ejemplo:

```
>(circula '((que) sucede aqui))
(sucede aqui (que))
```

Reescribe la función y llámala CIRCULA-DIR de tal manera que puedas desplazar la lista en ambas direcciones. Esto es, deberá funcionar como sigue:

```
>(circula-dir '(1 2 3 4) 'izq)
(4 1 2 3)

>(circula-dir '(1 2 3 4) 'der)
(2 3 4 1)
```

3.- Un palíndromo es una secuencia de caracteres que se leen igual al derecho que al revés. La función PALINDROMEP siguiente verifica si una lista es un palíndromo.

```
(defun palindromop (lst)
  (equal lst (reverse lst)) )
```

Por ejemplo:

```
>(palindromop '(1 2 3 4 5 4 3 2 1))
T
>(palindromop '(a b b a))
T
>(palindromop '(1 2 3))
NIL
```

Escribe una versión recursiva de esta función y llámala R-PALINDROMOP sin usar la función `reverse`.