**TC2006 – Lenguajes de Programación***Funciones de Orden Superior en Scheme*

Matrícula1\_\_\_\_\_\_\_\_\_\_\_ Matrícula2 \_\_\_\_\_\_\_\_\_\_\_\_

Resuelve los siguientes problemas SIN utilizar funciones de orden superior. Debes de subir a blackboard el archivo fuente (texto) que contenga todas las funciones.

1. Programar la función recursiva **aplica-listas** que aplique una lista de funciones binarias a cada elemento correspondiente en dos listas del mismo tamaño para obtener una lista de sublistas con los resultados de cada operador.

Probar con:

> (aplica-listas ‘(+ - \* /) ‘(1 2 3) ‘(4 5 6))

=> ((5 7 9)(-3 -3 -3)(4 10 18)(1/4 2/5 1/2))

> (aplica-listas ‘(cons list append) ‘((a b)) ‘((c d)))

=> ((((a b) c d))(((a b)(c d)))((a b c d)))

1. Programar el predicado recursivo **alguno?** que verifique si al menos un par elementos correspondientes de dos listas cumplen con un predicado binario.

Probar con:

> (alguno? < ‘(9 2 10) ‘(6 7 8)) => #t

> (alguno? (lambda (x y) (negative? (- x y)) ‘(5 8 10) ‘(4 5 6))

=> #f

> (alguno? > ‘(3 1 15) ‘(6 7 8)) => #t

1. Programar la función **verbosa** que genere un procedimiento unario que despliegue el valor de su argumento y su resultado además de regresarlo.

Probar con:

> (define vinc (verbosa (lambda (x) (+ x 1))))

> (vinc 5)

=> Entrada = 5

Salida = 6

6  
> (define vcdr (verbosa cdr))

> (vcdr ‘(1 2 3))

=> Entrada = (1 2 3)

Salida = (2 3)

(2 3)