

1. Evalúa **paso a paso** las siguientes expresiones: **(5 puntos)**

- i. `(equal? '(((() . c) . b) . a) (reverse (append '(a) '(b) '(c))))`
- ii. `((curry filter) list?) '(a b c))`
- iii. `((lambda(x . y) (append x y)) '(a b) '(c d) '(e f))`
- iv. `(let ((x 1) (y 2))
 (let* ((z x) (x y) (y (+ x 1))) (list x y z)))`
- v. `(map (car (list cons 5)) '(1 2) '(3 4))`

2. Responde **justificadamente** a las siguientes cuestiones: **(5 puntos)**

- i. ¿Cuándo una función se considera de orden superior?
- ii. ¿Analiza y justifica la siguiente afirmación: “la FOS filter de Scheme puede considerarse un *patrón de programación*”.
- iii. ¿Es cierto que las S-expresiones o bien son listas o bien son átomos?
- iv. ¿Qué es la evaluación perezosa y que ventaja tiene frente a la evaluación ansiosa? Pon un ejemplo.
- v. ¿En qué consiste la técnica de Curryng? Pon un ejemplo.