

1. (define test1
2. '(let ((m (module
 - a. (define twice (lambda (f)
 1. (lambda (x)
 - a. (f (f x)))))))))
3. (with (twice) m
 - a. (if ((twice not?) #f)
 - i. ((twice (lambda (x) (+ 1 x))) 4)
 - ii. 5)))
4. (define test2
5. '(let ((m (module
 - a. (define a 4)
 - b. (define b 5)))
 - c. (b 6)))))
6. (with (a) m
 - a. b)))