(1) (define x!
(lambda (x)
(if (= x 1) 1
(* x (x! (- x 1))))...)

(define fact x!)
(define x! (lambda (x) x))
(fact 5)

a): 20 b) 120 c) ...

(2) g x = x : (g x)

a) Calculul nu se termină.
b) (1)
c) (1)—(1)— ... ∞

(3) (define f
(lambda ()
(+ 1 (call/cc (lambda (c)
(g c))) 3))))

(define ~~gfact x!~~)
(lambda (c) (c (c (c 10)))))

(f)

a) 4 b) 14 c) eroare.

(4) Forma normală a expresiei este:
$(\lambda x.\, y \quad (\lambda x.(* \; x) \quad \lambda x.(* \; x)))$

a) $\lambda x.y$ b) nu are forma normală c) y.

(5) if TRUE then 2 else "0" : []

(a) Integer b) eroare de tip. c) [String]

(6) Închiderea funcțională. - definiție

(7) $f(x) = x + x + x$

De câte ori se evaluează x la apelul funcției

a) 3 b) 0 c) 1

(8) $\forall x, y, z . (x \Rightarrow y) \wedge (y \Rightarrow z) \Rightarrow (x \Rightarrow z)$?

a) af. e falsă b) satisfiabilă c) validă

(9) a) ((f x y) z)

b) . . . .

(a) (define f
(lambda (x y)
(lambda (z) ...)...)

~~Care e apelul funcției?~~

(b) . . .

(c) . . .

(10) (deffacts facts (f 0))
(defrule r_1
(logical (f ?x & : (< ?x 10)))
=>
(assert (f (+ 1 ?x))))
(defrule r_2
?f ← (f 1) (f 10)
=>
(retract ?f))

Ce fact-uri rămân

(a) niciuc b) (inițial-fact) (f 0)

(c) inițial-fact.

(11) Lant Markov

Registru SR : 1121111

M ({ 1, 2, - })

1 : 1R1 → 2 ;

2 : 21 → 2-1 ;

3 : 2 → . ;

end.

Registrul SR va contine:

a) -1 b) 2 c) 2-1

(12) (define p
(let* ((x 1) (p (delay
(begin (set! x (+ x 1))
x))))
p))
(define x 9)
(define y (+ (force p) (force p)))

Ce val. va avea y.

a) 4 b) 5 c) 20