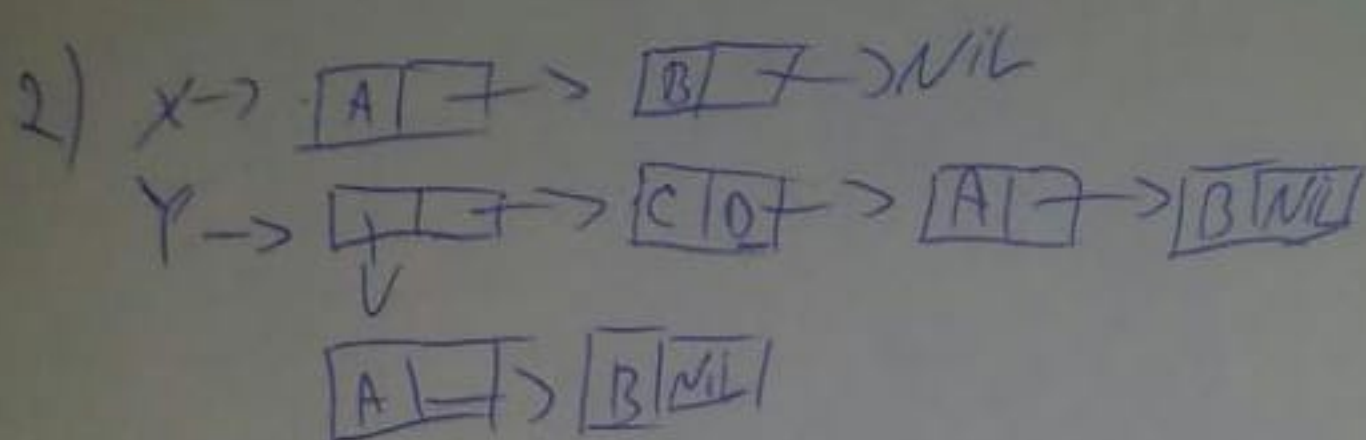
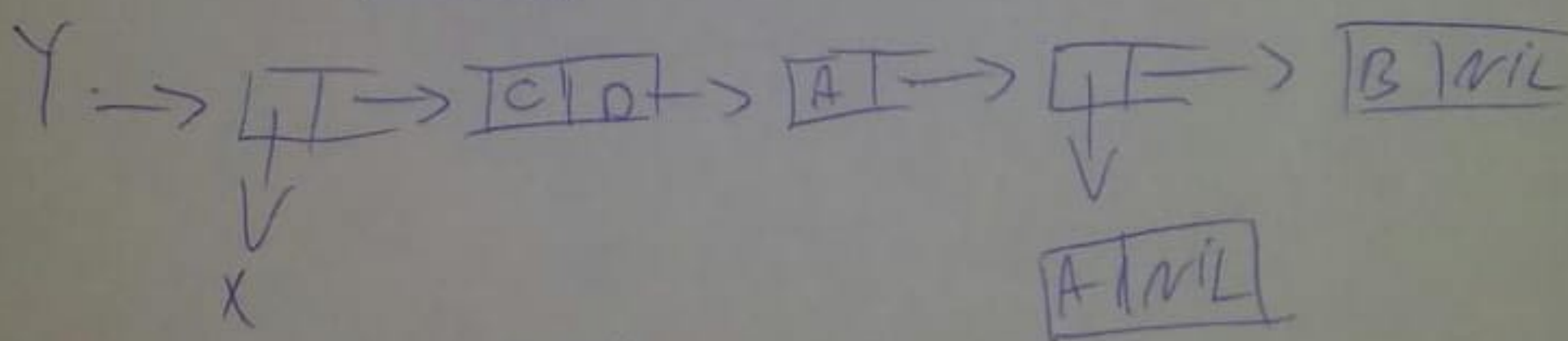
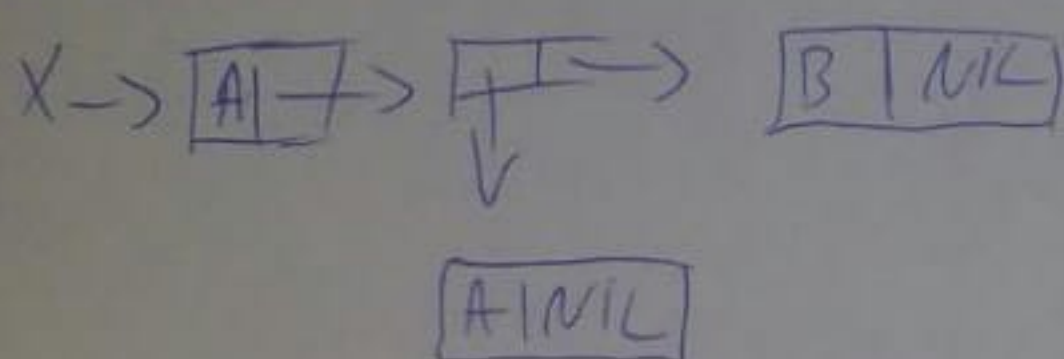


Partial

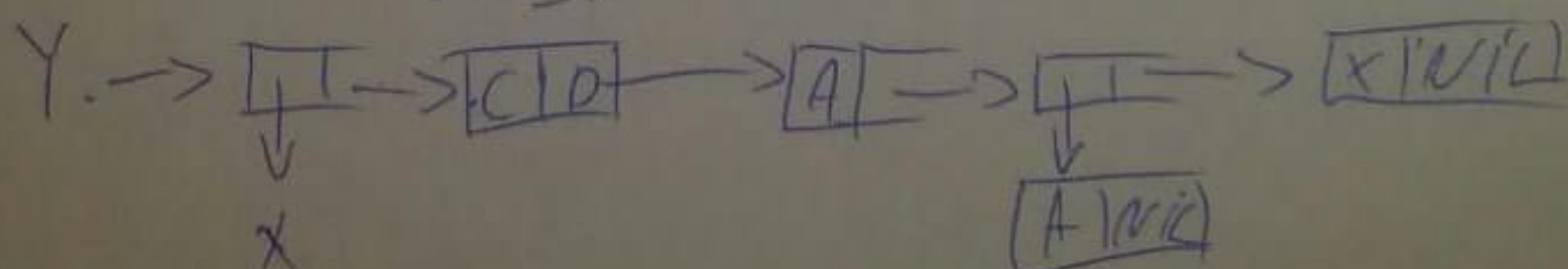
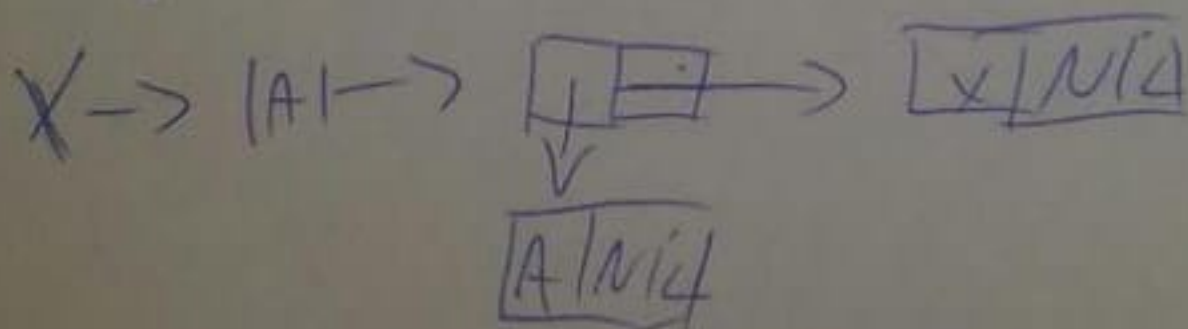
- 1) eg - Dacă o și b sunt exact același obiect (eg (cons 'a 'b) (cons 'a 'b))
 Equal - Dacă rezultatele lor eodax. (eg (cons 'a 'b) (cons 'a 'b))
 TRUE FALSE



Rplacd x. -



Setf (nth 2 x) 'x)



- 1) lambda → o funcție simplă și nu mai e folosit altundeva în program.
 (lambda (a b ...) (de executat cu parametri) valorile parametrilor)
 Ex: (lambda (a b) (* a b) 3) ⇒ 27
 - Înmulțește a și b și returnează valoarea.

3) Prima definieste un macro care reprezintă o bucată de cod
 (if T ≠ 8) se expandează în (if T ≠ 8) care rezultă 7
 (if N ≠ 8) ... rezultă 8. etc fiind evaluate

A doua definieste o funcție care primește ca parametru 3 obiecte (a b c)
 și returnează b sau c în funcție de a

f("char" "bchar" "character") va returna bchar.

4) a) (defun 4a (l)

~~(if (numberp (car l)~~
 (if (endp l) 1
 (if (numberp (car l))
 (* (car l) (4a (cdr l)))
 (4a (cdr l))

b) (defun 4b (l)
 4bh (l 1)

(defun 4bh (l p)

(if (endp l) 1
 (if (numberp (car l))
 (4bh (cdr l) (* p (car l)))
 (4bh (cdr l) p)

4) c) (defun 4c (l)

(do

(

(l l (cdr l))

(p (if (numberp (car l)) (* p (car l)) 1p)))

)

(

if (endp l) p

)

k

)

)

4) d) (defmacro 4d (l) '(

4) e) (defun 4e (l)

4eh (l 1)

)

(defun 4eh (l p)

(if (endp l) p

(if (numberp (car l))

(4eh (cdr l) (* p (car l)))

(if (not (atom (car l)))

(4eh (car l) p)

(4eh (cdr l) p)

)

)

)

)

5) Cu 0 param. x este NIL, este atom deci y este tot NIL
rezultă NIL adică Y

Cu 1 param:

$X \rightarrow \text{Atom} \Rightarrow Y \text{ NIL}$ N rezultă NIL adică Y
 $X \rightarrow \text{Listă} \Rightarrow Y$ este (caz X) și caută cel mai mare număr din X

Cu 2 param:

$X \rightarrow \text{Atom} \Rightarrow$ rezultă Y

$X \rightarrow \text{Listă} \Rightarrow$ caută și returnează cel mai mare nr din X
dacă acest număr este mai mare decât Y ,
 Y altfel.

Erori apar în momentul în care funcția primește ca parametru

o listă în care apare și un atom care nu este număr.

7) Prima funcție însumează elementele doar lui X fiindcă
apply-ul este efectuat pe X nu pe rezultatul din apply
 \Rightarrow incorrect

A doua funcție însumează ambele liste fiindcă nu se
modifică prima listă.
 \Rightarrow corect.

~~La ambele funcții~~

În prima funcție nu dă eroare dacă în X apare nil

În a doua funcție apar erori dacă nil apare în oricare
din cele două liste fiindcă NIL nu este un număr.