**Seminar 6 – Functii MAP**

(DEFUN triple (x)

(\* 3 x)

)

‘(1 2 3) => (3 6 9)

(mapcar #’triple ‘(1 2 3)) => (3 6 9)

<=> (list (triple 1) (triple 2) (triple 3))

‘(1 a 2 b 3) => (3 a 6 b 9)

(mapcar #’triple ‘(1 a 2 b 3)) => eroare

(DEFUN triple (x)

(cond

(( numberp x) (\* 3 x) )

(t x)

)

)

(mapcar #’triple ‘(1 a 2 b 3)) => (3 a 6 b 9)

(mapcar #’triple ‘(1 (a (2)) b 3)) => (3 (a (2)) b 9)

=> (3 (a (6)) b 9)

(DEFUN triple (x)

(cond

(( numberp x) (\* 3 x) )

((listp x) ( mapcar #’triple x))

(t x)

)

)

(mapcar #’triple ‘(1 (a (2)) b 3)) => (3 (a (6)) b 9)

<=> (triple ‘(1 (a (2)) b 3)) => (3 (a (6)) b 9)

(list (triple 1) (triple (a (2))) (triple b) (triple 3)) => (3 (a (6)) b 9)

3 (list (triple a) (triple (2)) b 9

(list a (list (triple 2)) ) => (a (6))

6

Triple(a)={ 3\*a ,daca a este numar

{ U I=1,n, triple(ai), daca a este o lista, unde a=(a1,…,an)

{ a, altfel

Pb1. Se da o lista neliniara.

Se cere sa se calculeze produsul tuturor numerelor din lista data.

Exemplu: ((2) 1 a (b 3)) => 6

Produs(a)= { a, a este atom numeric

{1,a este atom nenumeric

{ \prod I=1,n produs(ai), altfel unde a=(a1...an)

( defun Produs (a)

(cond

( (numberp a) a)

( (atom a) 1)

( t ( apply ‘\*( mapcar #’produs a))

)

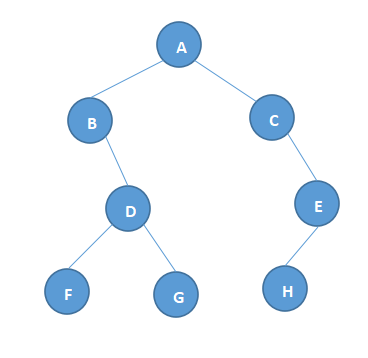
)

Pb2. Se consider un arbore n-ar reprezentat sub forma (rad (subarb1) (subarb2) … (subarbn)).

Se cere sa se calculeze numarul total de noduri aflate pe niveluri pare.

Se considera ca radacina este pe nivelul 1.

Exemplu:



(A (B () (D (F) (G))) (C () (E (H))))

=> 5

Noduri(a, n) = {0 ,(daca n % 2 = 1 si a este atom) sau a este ()

{1 , daca n % 2=0 si a este atom

{**Σ** ( I=1,n noduri(aI,n+1) , unde a=(a1 a2 … an)

(defun Noduri( a n)

(cond

((or (and (atom a) (= (mod n 2) 1)) (null a)) 0)

((and (atom a) (= (mod n 2) 0)) 1)

(t (apply ‘+ (mapcar #’Noduri a (+ 1 n))))

)

) => eroare

(defun Noduri( a n)

(cond

((or (and (atom a) (= (mod n 2) 1)) (null a)) 0)

((atom a) 1)

(t (apply ‘+ (mapcar #’(lambda (arb) (noduri arb (+ 1 n))) a )))

)

)