

1. Se dă o listă de liste. Se cere să se determine

(*lista* '((5 a (2 b (8))) (7 a (9)) (c d (10))))-->

```
(defun m (L)
  (cond
    ((numberp L) L)
    ((atom L) most-negative-fixnum)
    (t (apply #'max
              (mapcar #'m L)
              )
      )
  )
)

(defun lista (L)
  (mapcan #'(lambda (v)
              (cond
                ((= 0 (mod v 2)) (list v))
                (t nil)
              )
            ) (m L)
          )
  L)
)
```

2. Se dă o listă liniară. Se cere....

(*p* '(1 2 3)) ->

```
(defun p (L)
  (mapcan #'(lambda (e1)
              (mapcar #'(lambda (e2)
                          (list e1 e2)
                        )
                    L
                )
            )
  L)
)
```

3. O matrice se poate reprezenta sub forma unei liste formate din liste conținând elementele de pe linii, în ordine de la prima la ultima linie. De exemplu, lista ((1 2) (3 4)) corespunde matricei

$$\begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix}$$

Fie L o listă asociată unei matrici cu elemente numerice

```
(defun fct (L)
  (cond
    ((null (car L)) nil)
    (t (cons
         (mapcar #'car L)
         (fct (mapcar #'cdr L))
        )
      )
  )
)
```

$(fct '((1 2) (4 5) (7 8))) \rightarrow$

4. Se dă o mulțime reprezentată sub forma unei liste liniare. Se cere să se genereze lista submulțimilor mulțimii. Se va folosi o funcție MAP.

Ex: $(subm '(1 2)) \rightarrow (nil (1) (2) (1 2))$

```
(defun subm (L)
  (cond
    ((null L) (list nil))
    (t ((lambda (s)
          (append s (mapcar #'(lambda (sb)
                                (cons (car L) sb)
                                )
                )
          s
        )
        )
      )
    (subm (cdr L))
  )
)
```

5. Se dă o mulțime reprezentată sub forma unei liste liniare. Se cere să se genereze lista permutărilor mu lțimii. Se va folosi o funcție MAP.

Ex: (*permutari* '(1 2 3)) \rightarrow ((1 2 3) (1 3 2) (2 1 3) (2 3 1) (3 1 2) (3 2 1))

```
(defun permutari (L)
  (cond
    ((null (cdr L)) (list L))
    (t (mapcan #'(lambda (e)
                    (mapcar #'(lambda (p)
                                (cons e p)
                                )
                    (permutari (remove e L))
                    )
              )
      L
      )
    )
  )
)
```

6. Să se scrie o macrodefiniție (*dec x*) care să aibă următorul efect

```
(setq x 10)
(dec x)  $\rightarrow$  9
x  $\rightarrow$  9
```

```
(defmacro dec (n)
  (list 'setq n (list '- n 1))
  ; `(setq ,n (- ,n 1))
)
```