

1. Se dă o listă neliniară. Se cere să se șteargă din listă atomii numerici negativi. Se va folosi o funcție MAP.

Ex: (stergere '(a 2 (b -4 (c -6)) -1)) → (a 2 (b (c)))

```
(defun sterg(L)
  (cond
    ((and (numberp L) (minusp L)) nil)
    ((atom L) (list L))
    (t (list (apply #'append
                    (mapcar #'sterg L)

```

```

                    )
                )
            )
        )
    )

```

```
(defun stergere(L)
  (car (sterg L))
)
```

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

Ex: (permutari '(1 2 3)) → ((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
    )
  )
)
```

3. 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))
                                (subm (cdr L))
                                )
          )
        s)
      )
    )
  )
)
```

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

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

```
(defmacro dec(x)
  `(list 'setq x (list '- x 1))
  `(setq ,x (- ,x 1))
)
```