1. Funcția G returnează suma primelor 2 nr. dintr-o listă dată ca parametru.

(set f 'f) -> în acest caz, și f se evaluează, însă nu se știe la ce se poate evalua =>
=> fals

Pentru a fi corect există 2 soluții: (set 'f 'f)
(setq f 'f) ; în

ambele cazuri, f se evaluează la f
(continuarea problemei se bazează pe schimbarea de mai sus)

(set f 'G) -> f se evaluează la F
F se evaluează la G

(F '(2 3 4 5 6)) -> va da eroare, deoarece programul nu cunoaște o funcție cu numele F
(f se evaluează la g, dar și g trebuia evaluat pentru a se apela funcția)

O soluție în acest caz ar putea fi: (funcall f '(2 3 4 5 6))
(funcall îl evaluează pe f la g și apoi apelează funcția)

3. Lista submulțimilor (cu cel puțin un element) având suma div. cu 3.

Modele matematice:

candidat $(e_1 \dots e_m)$ =
1. e_1, $m \geq 1$
2. candidat $(e_2 \dots e_m)$, $m \geq 1$

$$\text{subm}(l, sum, col) = \begin{cases} col, & sum \% 3 = 0 \\ \text{subm}(l, sum+e, e \oplus col), & \text{dacă } e < col_1 \ (col = col_1 \dots col_m), \\ & \text{unde } e = \text{candidat}(l) \end{cases}$$

$$\text{submultime}(l) = \text{subm}(l, e, (e)), \text{ unde } e = \text{candidat}(l)$$

$$\text{main}(l) = \bigcup \text{submultime}(l)$$

```
candidat([H|_], H).
candidat([_|T], R):-
    candidat(T, R).
```

model de flux: (i, o)
nedeterminist

```
subm(_, Sum, Col, Col):- Sum mod 3 =:= 0.
subm(L, Sum, [H|T], R):-
    candidat(L, E),
    E < H,
    Sum1 is Sum + E,
    subm(L, Sum1, [E|[H|T]], R).
```

model de flux: (i, i, i, o)
nedeterminist

```
submultime(L, R):-
    candidat(L, E),
    subm(L, E, [E], R).
```

model de flux: (i, o)
nedeterminist

```
main(L, R):-
    findall(R1, submultime(L, R1), R).
```

model de flux: (i, o)
determinist

5. Nr. de subliste pt. care ultimul atom numeric este impar.

Modele matematice:

$$ultimul(l_1 \dots l_n, col) = \begin{cases} col_1, & n = 0 \ (col = col_1 \dots col_m) \\ ultimul(l_2 \dots l_n, l_1 \oplus col), & l_1 \text{ e nr.} \\ ultimul(l_2 \dots l_n, col), & l_1 \text{ e atom nonnumeric} \\ ultimul(l_2 \dots l_n, (ultimul(l_1, col))), & \text{altfel} \end{cases}$$

$$subliste(l) = \begin{cases} 0, & l \text{ e atom} \\ 1 + \sum_{i=1}^{n} subliste(l_i), & ultimul(l, \emptyset) \text{ este nr. impar} \\ \sum_{i=1}^{n} subliste(l_i), & \text{altfel} \end{cases}$$

```lisp
(defun ultimul (l col)
    (cond
        ((null l) (car col))
        ((numberp (car l)) (ultimul (cdr l) (cons (car l)
                                                  col)))
        ((atom (car l)) (ultimul (cdr l) col))
        (+ (ultimul (cdr l) (list (ultimul (car l) col))))
    )
)

(defun subliste (l)
    (cond
        ((atom l) 0)
        ((and (numberp (ultimul l nil)) (oddp (ultimul l nil)))
         (+ 1 (apply #'+ (mapcar #'subliste l))))
        (+ (apply #'+ (mapcar #'subliste l)))
    )
)
```