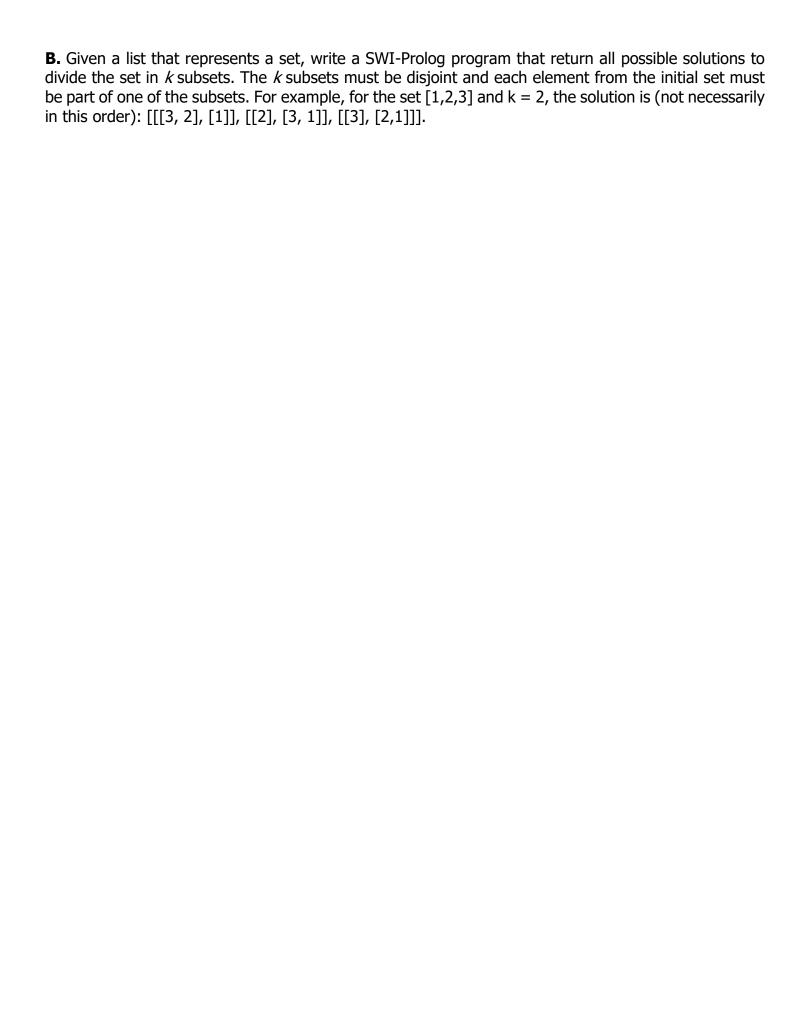
## Functional and logic programming written exam -

## **Important:**

- 1. Subjects are graded as follows: of 1p; A 1.5p; B 2.5p; C 2.5p; D 2.5p.
- 2. Prolog problems will be resolved using SWI Prolog. The following are required: (1) explanation of the code and of the reasoning behind it; (2) recursive model that solves the problem, for all the predicates used; (3) specification of every predicate (parameters and their meaning, flow model, type of the predicate deterministic/non-deterministic).
- 3. Lisp problems will be resolved using Common Lisp. The following are required: (1) explanation of the code and of the reasoning behind it; (2) recursive model that solves the problem, for each function used; (3) specification of every function (parameters and their meaning).
- A. Let G be LISP function and given the following definition (DEFUN F(L) (COND ((NULL L) 0) (> (G L) 2) (+(G L) (F (CDR L)))) (T (G L))

Rewrite the definition in order to avoid the repeated call **(G L)**. Do NOT redefine the function. Do NOT use SET, SETQ, SETF.Justify your answer.



**C.** Given a list composed of integer numbers, generate in PROLOG the list of arrangements of N elements ending with an odd value and have the sum S given. Write the mathematical models and flow models for the predicates used. For example, for the list L=[2,7,4,5,3], N=2 and S=7  $\Rightarrow$  [[2,5], [4,3]] (not necessarily in this order).

**D.** An n-ary tree is represented in Lisp as ( node subtree1 subtree2 ...). Write a Lisp function to verify whether a node x occurs on an even level of the tree. The root level is assumed zero. A MAP function shall be used.

**Example** for the tree (a (b (g)) (c (d (e)) (f)))

**a)** x=g => T **b)** x=h => NIL