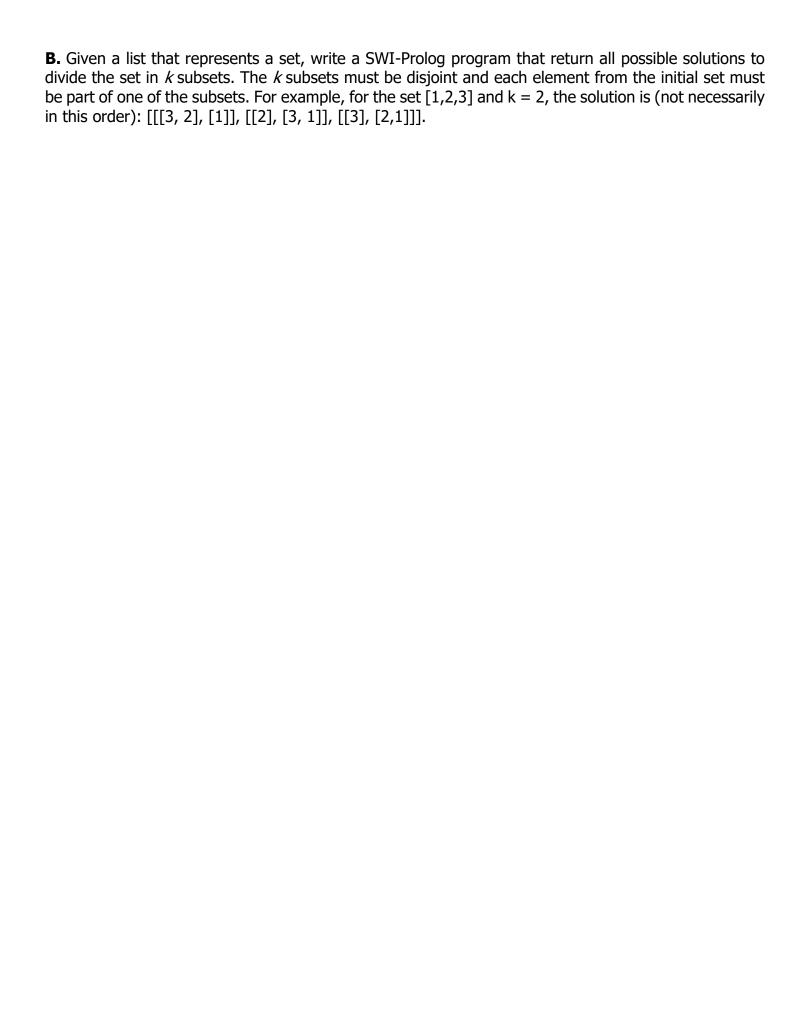
## Functional and logic programming written exam -

## **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.** The following function definition in LISP is given (DEFUN F(N) (COND ((= N 0) 0) (> (F (- N 1)) 1) (- N 2)) (T (+ (F (- N 1)) 1))

Rewrite the definition in order to avoid the double recursive call **(F (- N 1))**. Do NOT redefine the function. Do NOT use SET, SETQ, SETF. Justify your answer.



**C.** Write a PROLOG program that determines from a list made of integer numbers, the list of subsets with at least 2 elements, composed of numbers in strictly increasing order. Write the mathematical models and flow models for the predicates used. For example for the list  $[1, 8, 6, 4] \Rightarrow [[1,8],[1,6],[1,4],[6,8],[4,6],[1,4,6],[1,4,8],[1,6,8],[4,6,8],[1,4,6,8]]$  (not necessarily in this order).

**D.** An n-ary tree is represented in Lisp as ( node subtree1 subtree2 ...). Write a Lisp program to return the *height* of a node of a tree. **A MAP function shall be used.** 

**Example** for the tree (a (b (g)) (c (d (e)) (f))) **a)** nod=e => the height is 0 **b)** nod=v => the height is -1

c) nod=c => the height is 2.