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))
    )
)
```

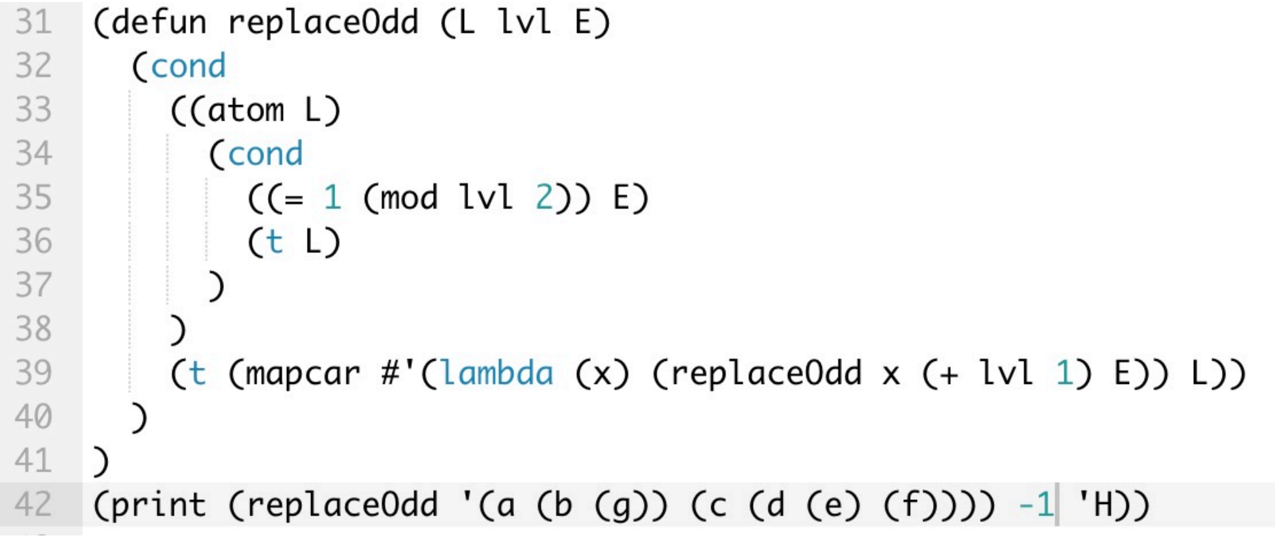
Rewrite the definition in order to avoid the repeated call **(G L)**. Do NOT redefine the function. Do NOT use SET, SETQ, SETF.Justify your answer.

B. Given a list composed of only sublists that contain positive digits, write a SWI-Prolog program that computes the greatest even number that can be formed picking one digit from each sublist. The digits in the resulted number must be in the same order as in the sublists they come from. Each sublist will contain at least one even digit. For example, for the list [[2,5,1,9], [7,2,1], [9,4,6,5], [2,6,0,7]] the result will be 9796.

C. Write a PROLOG program that generates the list of all combinations of k elements with the value of sum of each combination even number, from a list of integers. Write the mathematical models and flow models for the predicates used. For example, for the list L[6, 5, 3, 4], **k**=2 ⇒ [[6,4],[5,3]] (not necessarily in this order).

D. An n-ary tree is represented in Lisp as (node subtree1 subtree2 ...). Write a function to replace all nodes on odd levels with a given value **e**. The root level is assumed zero. **A MAP function shall be used.**

Example for the tree (a (b (g)) (c (d (e)) (f))) and **e**=h => (a (h (g)) (h (d (h)) (h)))



```
(defun replaceOdd (L lvl E)
  (cond
    ((atom L)
      (cond
        ((= 1 (mod lvl 2)) E)
        (t L)
      )
    )
    (t (mapcar #'(lambda (x) (replaceOdd x (+ lvl 1) E)) L))
  )
)
(print (replaceOdd '(a (b (g)) (c (d (e) (f)))) -1 'H))
```