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(L)
    (COND
        ((ATOM L) -1)
        ((> (F (CAR L)) 0) (+ (CAR L) (F (CAR L)) (F (CDR L))))
        (T (F (CDR L)))
    )
)
```

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

B. Given a heterogeneous list made of numbers and nonempty numeric lists, write a SWI-PROLOG program that verifies if all numbers (including those in sublists) form an increasing sequence of numbers. For example, for the list [2,4,6, [10, 12, 19], 30, 201, [1000, 1003, 1006, 2003], 2020] the result will be true, but for the list [2,4,6, [10, 12, 11], 30, 201, [1000, 1003, 1006, 2003], 2020] the result will be false.

C. Write a PROLOG program that generates the list of all subsets with at least N elements such that the value of sum of all elements from each subset is divisible with 3, from a list of integers. Write the mathematical models and flow models for the predicates used. For example, for the list L=[2,3,4] and **N**=1 ⇒ [[3],[2,4],[2,3,4]] (not necessarily in this order).

```prolog
83  subS([], []).
84  subS([_|T], L):-
85       subS(T, L).
86  subS([H|T], [H|L]):-
87       subS(T, L).
88
89  suma([], 0).
90  suma([H|T], R):-
91       suma(T, R1),
92       R is R1 +H.
93
94  oneSol2(L, N, S):-
95       subS(L, Sub),
96       length(Sub, Len),
97       Len >= N,
98       suma(Sub, Sum),
99       Sum mod 3 =:= 0,
100      S = Sub.
```

D. An n-ary tree is represented in Lisp as (node subtree1 subtree2 ...). Write a Lisp function to replace all nodes on the given level **k** 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) k=2 => (a (b (h)) (c (h (e)) (h)))   **(b)** k=4 => (a (b (g)) (c (d (e)) (f)))

```lisp
1   (defun replaceK(L K E)
2     (cond
3       ((atom L)
4         (cond
5           ((= -1 K)  E)
6           (t L)
7         )
8       )
9       (t (mapcar #'(lambda (x) (replaceK x (- K 1) E)) L))
10    )
11  )
12
13  (print (replaceK '(a (b (g)) (c (d (e)) (f))) 2 'h))
```