DRAFT

(DEFUN F (G L)

~~10. Grading: (def) – 1p; (1) – 3p (4+6 ?)p; (2) – 3p; (3) – 3p.~~

~~*The Prolog problems will be solved in SWI Prolog. You will explain the code, give the reasoning, predicate specification including recursive formula, flow model, meaning of all variables and parameters. The Lisp problems will be solved in Common Lisp. You will explain the code, give the reasoning, function specification, meaning of all variables and parameters, the formula for recursion. The MAP problem implies writing a main and an auxiliary function. For a penalty, this may be solved without using MAP functions.*~~

1.

I.1 Consider the following function definition in LISP

```
(DEFUN F (G L)
    (COND
        ((NULL L) NIL)
        (> (FUNCALL G L) 0) (CONS (FUNCALL G L) (F (CDR L))))
        (T (FUNCALL G L))
    )
)
```

Rewrite it in order to have only one recursive call (FUNCALL G L). Do not create global variables. Do not write a new subalgorithm to achieve the same thing. Justify the answer.

I.2 Let L be a numerical list and consider the following PROLOG definition for the predicate with the flow model (i, o):

```
f([],-1).
f([H|T],S) :- f(T,S1),S1>0,!,S is S1+H.
f([_|T],S) :- f(T,S1),S is S1.
```

Rewrite the predicate in order to have only one recursive call f(T,S1) in all clauses. Do not write a new predicate to achieve the same thing. Justify the answer.

I.3 The LISP function G is defined by (DEFUN G(L) (LIST (CAR L) (CAR L))). In order to rename the function G we execute (SETQ Q 'G) followed by (SETQ P 'Q). What is the result of evaluating the form (FUNCALL (EVAL P) '(A B C))? Justify the answer.

*I.4 Let L be a numerical list and consider the following PROLOG definition for the predicate **g(list, list)** with the flow model (i, o):*

```
g([],[]).
g([_|T],S) :- !,g(T,S).
g([H|T],[H|S]) :- H mod 2 =:= 0,g(T,S).
```

Give the result of the following goal: **g([1,2,3],L).** Justify the answer.

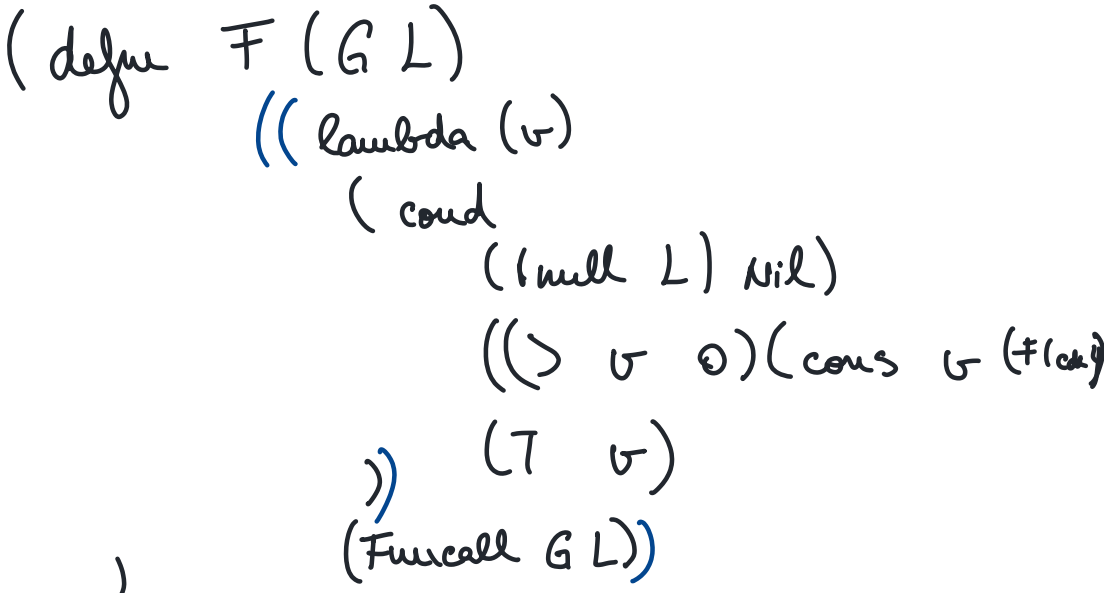
II. For a given value N, write a PROLOG program to generate the list of all permutations w elements N, N+1, ... ,2*N-1 with the property that absolute value of difference between t consecutive elements from each permutation is <=2. Write the mathematical model, flow mo and the meaning of all variables for each predicate used.

III. Given an n-ary tree represented as (root (subtree_1) (subtree_2)... (subtree_n)) in which n are numerical and non-numerical atoms, multiply each numerical node of the tree with its Root is on level 1. Use MAP functions. Write the mathematical model and the meaning parameters for each function used. Example 1: (1(2)(3(4(5)))) => (1 (4) (6 (12 (20)))) Example 2 : (1(2(A(B)))(3(4(C(5))))) => (1 (4 (A (B))) (6 (12 (C (25)))))

II.

```
4  insertE(E, L, [E|L]).
5  insertE(E, [H|L], [H|Res]):-
6      insertE(E, L, Res).
7
8  perm([], []).
9  perm([E|T], P):-
10     perm(T, L),
11     insertE(E, L, P). %(i,i,o)
12
13 validPerm([]).
14 validPerm([_]).
15 validPerm([H1,H2|T]):-
16     Diff is abs(H1-H2),
17     Diff =< 2,
18     validPerm([H2|T]).
19
20 findPerm(N, P):-
21     F is 2*N-1,
22     numlist(N, F, L),
23     perm(L, P),
24     validPerm(P).
25
26
27 main(N, P):-
28     findall(Res, findPerm(N, Res), P).
```

I1.



```
(defun multiplyLevel (L K)
  (cond
    ((atom L)
      (cond
        ((numberp L) (* L K))
        (T L)
      )
    )
    (T (mapcar #'(lambda (x) (multiplyLevel x (+ 1 K))) L))
  )
)

(print (multiplyLevel '(1 (2) (3 (4 (5)))) 0))
(print (multiplyLevel '(1 (2 (A (B))) (3 (4 (C (5))))) 0))
```