Functional and logic programming
- written exam -

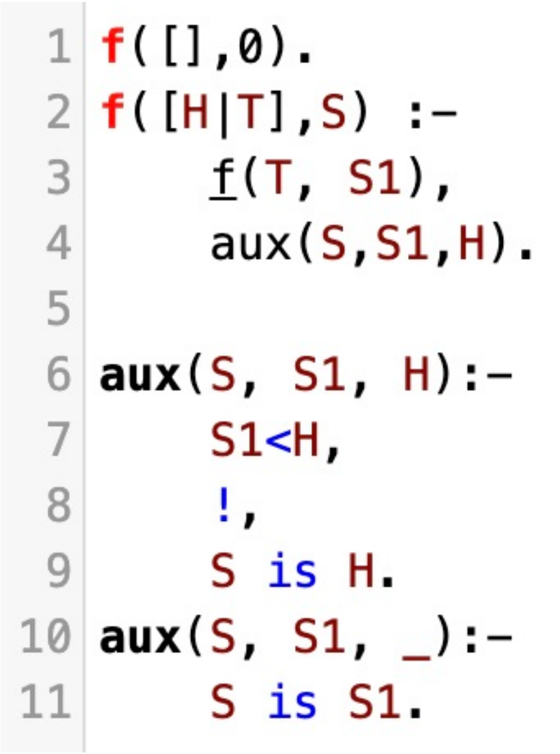
Important:

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 L be a list of numbers and given the following PROLOG predicate definition **f(list, integer)**, with the flow model (i, o):

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

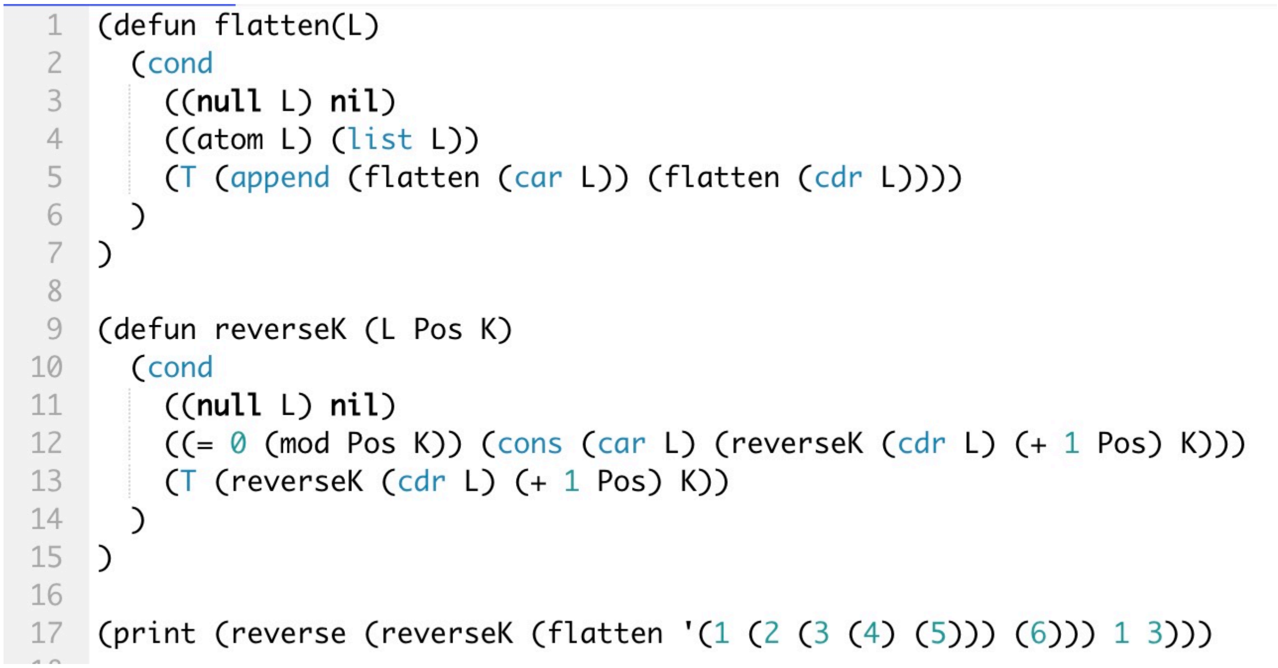
Rewrite the definition in order to avoid the recursive call **f(T,S)** in both clauses. Do NOT redefine the predicate. Justify your answer.



```
f([],0).
f([H|T],S) :-
    f(T, S1),
    aux(S,S1,H).

aux(S, S1, H):-
    S1<H,
    !,
    S is H.
aux(S, S1, _):-
    S is S1.
```

B. Given a nonlinear list containing both numerical and non-numerical atoms, write a Lisp program that builds a list with the elements from the initial list, from positions k to k (counting from left to right, considering all elements regardless of level), in reverse order. For example, for the list (A B 12 (5 D (A F (10 B) D (5 F) 1)) C 9) and k = 3 the result is (9 F B A 12).

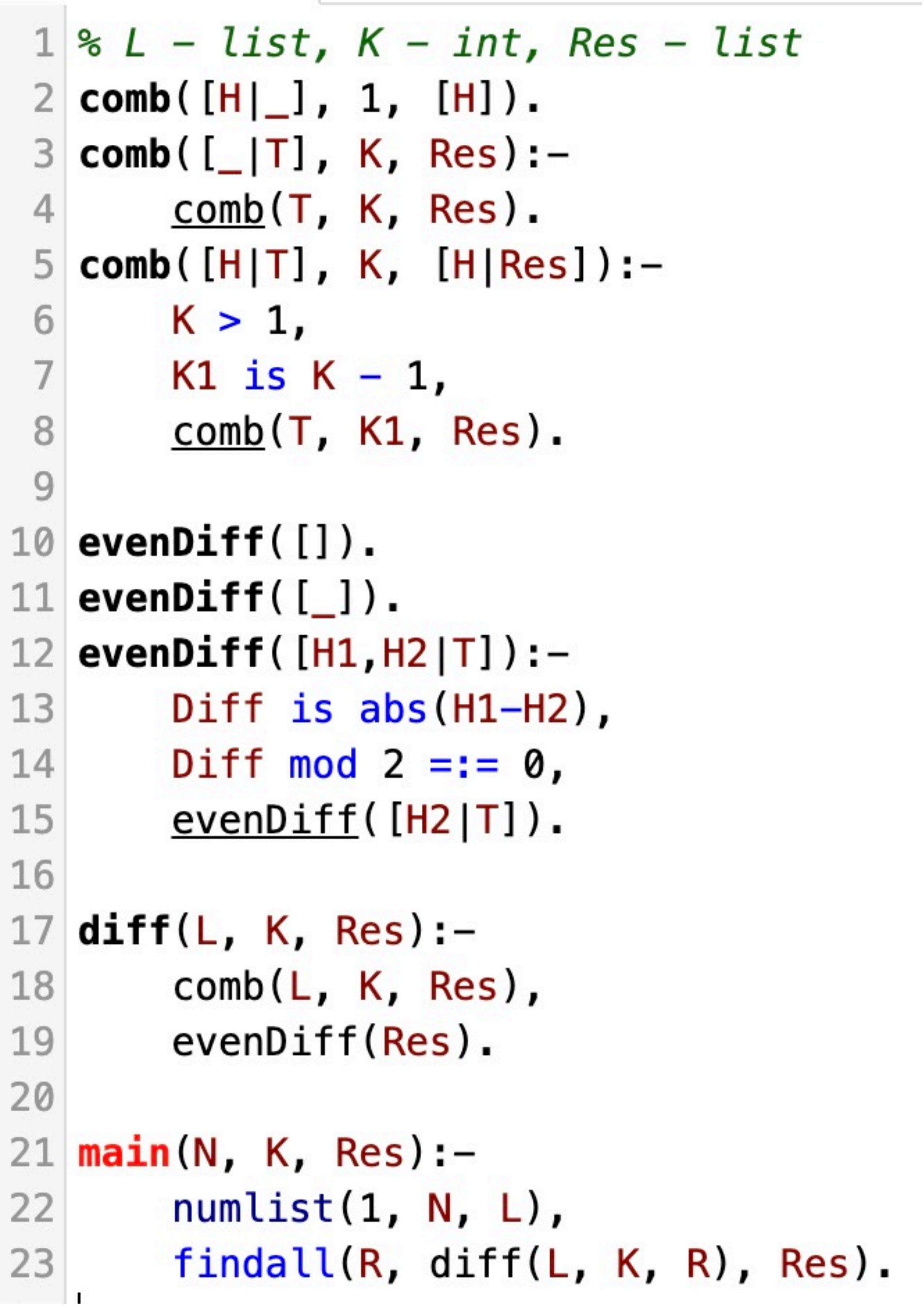


```
(defun flatten(L)
  (cond
    ((null L) nil)
    ((atom L) (list L))
    (T (append (flatten (car L)) (flatten (cdr L))))
  )
)

(defun reverseK (L Pos K)
  (cond
    ((null L) nil)
    ((= 0 (mod Pos K)) (cons (car L) (reverseK (cdr L) (+ 1 Pos) K)))
    (T (reverseK (cdr L) (+ 1 Pos) K))
  )
)

(print (reverse (reverseK (flatten '(1 (2 (3 (4) (5))) (6))) 1 3)))
```

C. Write a PROLOG program that generates the list of all combinations of k elements with numbers from 1 to N, with the property that difference between two consecutive numbers from a combination has an even value. Write the mathematical models and flow models for the predicates used. For example, for the **N**=4, **k**=2 ⇒ [[1,3],[2,4]] (not necessarily in this order).



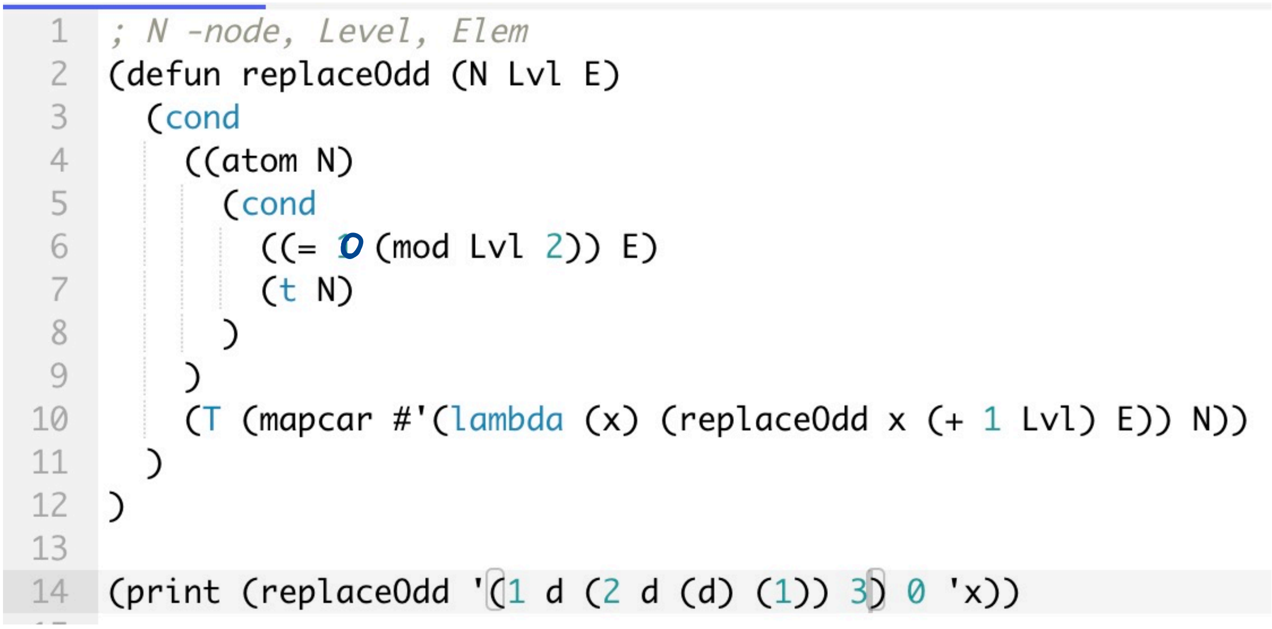
```
% L - list, K - int, Res - list
comb([H|_], 1, [H]).
comb([_|T], K, Res):-
    comb(T, K, Res).
comb([H|T], K, [H|Res]):-
    K > 1,
    K1 is K - 1,
    comb(T, K1, Res).

evenDiff([]).
evenDiff([_]).
evenDiff([H1,H2|T]):-
    Diff is abs(H1-H2),
    Diff mod 2 =:= 0,
    evenDiff([H2|T]).

diff(L, K, Res):-
    comb(L, K, Res),
    evenDiff(Res).

main(N, K, Res):-
    numlist(1, N, L),
    findall(R, diff(L, K, R), Res).
```

D. Given a nonlinear list, write a Lisp function to return the list with all atoms on even levels replaced by zero. The superficial level is assumed 1. **A MAP function shall be used.**
<u>Example</u> for the list (a (1 (2 b)) (c (d))) the result is (a (0 (2 b)) (0 (d))).



```lisp
; N -node, Level, Elem
(defun replaceOdd (N Lvl E)
  (cond
    ((atom N)
      (cond
        ((= 0 (mod Lvl 2)) E)
        (t N)
      )
    )
    (T (mapcar #'(lambda (x) (replaceOdd x (+ 1 Lvl) E)) N))
  )
)

(print (replaceOdd '(1 d (2 d (d) (1)) 3) 0 'x))
```