# Examen PP – Seria CC

16.06.2017

Timp de lucru 2 ore . 100p necesare pentru nota maximă

---

1. Determinați forma normală pentru următoarea expresie, ilustrând pașii de reducere:
   $((\lambda x.\lambda y.\lambda z.(x\ \ y)\ \ \lambda x.x)\ \ z)$ 15p

2. Este vreo diferență (ca efect, la execuție) între cele două linii de cod Racket? Dacă da, care este diferența?; dacă nu, de ce nu diferă?
   ```
   (let ((a 1)) (let ((b a)) (+ a b)))
   (let* ((a 1) (b a)) (+ a b))
   ```
   15p

3. Implementați în Racket funcția `f` care primește o listă și determină cel mai mare element. Folosiți, în mod obligatoriu, cel puțin o funcțională. 15p

4. Sintetizați tipul funcției `f` (în Haskell): `f g h l = map (g . h) l` 15p

5. Scrieți definiția în Haskell a clasei `Ended` care, pentru un tip colecție `t` construit peste un alt tip `v`, definește o funcție `frontEnd` care extrage primul element din colecție și o funcție `backEnd` care extrage ultimul element din colecție.
   Instanțiați această clasă pentru tipul listă Haskell. 15p

6. Știind că *Cine spune multe, spune mai puțin decât cine tace* , și că $spune\_multe(Ion)$ și $tace(Marcu)$ , demonstrați folosind rezoluția că $spune\_mai\_putin(Ion, Marcu)$ este adevărat . 15p

7. Implementați în Prolog predicatul `x(L, A, B, N)` care detemină, pentru o listă `L`, numărul `N` de elemente care sunt mai mari decât `A` și mai mici decât `B`. Nu folosiți recursivitate explicită. 15p

8. Implementați un algorim Markov care primește în șirul de intrare un număr binar și adună 1 la acest număr. Exemple: 0 + 1 = 1; 1 + 1 = 10; 11 + 1 = 100; 100 + 1 = 101 15p

A