

## Examen PP – Seria CD

14.06.2017

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

---

1. Determinați forma normală pentru următoarea expresie, ilustrând pașii de reducere:  
 $((\lambda x.\lambda y.\lambda z.y \ \Omega) \ \lambda x.(x \ z))$  10p
2. Ce întoarce următorul cod Racket și ce reprezintă valoarea întoarsă?  
`(let f ((n 0) (m 5)) (if (< n m) (+ n (f (+ n 1) m)) 0))` 10p
3. De câte ori se evaluează expresia (E 1) în codul Racket de mai jos?  
`(let ((proc (lambda (x) (+ (force x) (force x)))) (proc (delay (E 1))))`  
Tranformați codul dat pentru a folosi închideri.  
Este vreo diferență în execuție? 20p
4. Implementați în Haskell o funcție care primește 3 liste infinite și le adună element cu element. E.g. `f [1..] [4..] [7..] = [12, 15..]` 15p
5. Se dă clasa Haskell: `class Summable t where sum :: (Num a) => t a -> a`,  
cu semnificația că funcția din clasă adună toate elementele conținute în argument.
  - Instanțiați clasa dată pentru tipul listă;
  - Definiți un tip *listă imbricată de elemente de tip a*;
  - Instanțiați clasa dată pentru tipul definit.20p
6. Știind că *Vulpea care doarme nu prinde găini*, și că *vulpe(Vicky), gaina(Gini), și doarme(Vicky)*, demonstrați folosind rezoluția că *prinde(Vicky, Gini)* este fals . 15p
7. Implementați în Prolog un predicatul `p(L1, L2, L3)`, care primește în L1 și L2 două liste de lungimi egale și leagă L3 la o listă de perechi, fiecare pereche conținând elementele de la același index din L1 și L2 (similar `zip` din Haskell) . Exemplu: este adevărat `p([1, 2, 3], [a, b, c], [(1, a), (2, b), (3, c)])`. 15p
8. Implementați un algoritm Markov care primește un șir format din simboluri 0 și 1 și verifică dacă șirul începe și se termină cu același simbol. Dacă simbolurile de la început și de la sfârșit sunt identice, scrie la sfârșitul șirului simbolurile "ok", altfel nu schimbă șirul cu nimic. Încercați să scrieți algoritmul pentru un alfabet de bază A oarecare. 15p