

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:
 $((\lambda x.\lambda y.\lambda z.(x\ y)\ \lambda x.y)\ a)$ 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?

```
(define a 2) (let ((c 2)) (let ((a 1) (b a)) (+ a b)))
```

```
(define a 2) (let* ((c 2) (a 1) (b a)) (+ a b))
```

 15p
3. Implementați în Racket funcția `f` care primește o listă și determină elementul cu cel mai mare modul. Folosiți, în mod obligatoriu, cel puțin o funcțională. 15p
4. Sintetizați tipul funcției `f` (în Haskell): `f x y z g = map g [x, y, z]` 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 `data NestedL a = A a | L [NestedL a]` 15p
6. Știind că *Un bogat când moare, săracul fluiere*, și că *bogat(Bill), sarac(Bob), și moare(Bill)*, demonstrați folosind rezoluția că *fluiere(Bob)* este adevărat . 15p
7. Implementați în Prolog predicatul `x(L, M)` care determină, pentru o listă `L`, `M`, maximul listei. Nu folosiți recursivitate explicită. 15p
8. Implementați un algoritm Markov care primește în șirul de intrare un număr binar și adună 2 la acest număr. Exemple: `1 + 10 = 10`; `10 + 10 = 100`; `1000 + 10 = 1010`; `101 + 10 = 111`; `111 + 10 = 1001` 15p