

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

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

 15p
3. Implementați în Racket funcția `f` care primește o listă și determină cel mai mic element. Folosiți, în mod obligatoriu, cel puțin o funcțională. 15p
4. Sintetizați tipul funcției `f` (în Haskell): `f x y z g = filter 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 Pair a = MakePair a a` 15p
6. Știind că *Un om cum își așterne, așa doarme*, și că *asterne(Nectarie,bine)* și *om(Nectarie)*, demonstrați folosind rezoluția că *doarme(Nectarie,bine)* este adevărat . 15p
7. Implementați în Prolog predicatul `x(L, A, B, SL)` care determină, pentru o listă `L`, numărul `N` de elemente care nu sunt mai mari decât `A` și mai mici decât `B`. Nu folosiți recursivitate explicită. 15p
8. Implementați un algoritm Markov care primește în șirul de intrare un număr binar și scade 1 din acest număr. Exemple: `1 - 1 = 0`; `10 - 1 = 1`; `11 - 1 = 10`; `100 - 1 = 11`; `1010 - 1 = 1001` Este ok dacă numărul rezultat începe cu 0. 15p