Facultatea de Informatică Numele: ……Anul……

Test scris la Programare funcțională

2 Iunie 2011

Observaţii: *Toate întrebările sunt obligatorii. Întrebările 1-6 sunt notate cu câte 4 puncte iar subiectele 7-10 cu câte 6 puncte. Se acordă 5 puncte din oficiu.*

1. Explicați termenul "tuplă (n-uplă)" în Haskell. Dați 3 exemple de funcții în care apar tuple.
2. Ce reprezintă construcția: **`data Bin = Zero|Unu`**
   a. Definește Bin ca fiind sinonim cu tipul Int si valori **`Zero|Unu`**
   b. Definește expresia regulata **`Zero|Unu`**
   c. Definește Bin ca fiind un nou tip de data cu valorile **`Zero|Unu`**
   d. Definește valoarea **`Bin`** ca având alternativ valori Zero sau Unu
3. Expresia **`\ f -> (\ g -> g.f)`** are tipul :
   a. **`(a -> b) -> (b -> c) -> a -> c`**
   b. **`(a -> b) -> (a -> c) -> b -> c`**
   c. **`(a -> b) -> (b -> a) -> a`**
   d. **`(a -> b -> c) -> a -> c`**
4. Care este rezultatul execuției `map(\x->x`mod`3) [z|z<-[2,6..20]]` ?
   a. `2`
   b. `[2,0,1,2,0]`
   c. `Se raportează eroare`
   d. `[0,2,3,4,6]`
5. La declararea unui tip ca instanță a clasei Monad trebuiesc definite :
   a. return și operatorul >>
   b. return și operatorul >>=
   c. operatorii >>= și >>
   d. doar return
6. In declarația **`module TipNou(…) where`**…, în paranteză apar despărțite prin virgula :
   a. Constructorii noului tip
   b. Lista tuturor funcțiilor ce se declara în modulul respectiv
   c. Numele tipului urmat de cel al operațiilor ce pot fi aplicate valorilor acestui tip
   d. Nimic pentru ca sintaxa este alta

7. Să se definească funcția `count` care să determine numărul de apariții ale unui caracter într-un text. De exemplu `count 'a' "facultatea de informatica" = 5`.

8. Să se definească funcția `dup1: [a] -> [a]` care duplică fiecare componentă de indice impar dintr-o listă. De exemplu: `dup1 [1,2,3,4,5,6,7,8] = [1,2,2,3,4,4,5,6,6,7,8,8]`, `dup1 "alpha" = "allphha"`

9. Definiți funcția de concatenare a două liste `(++) :: [a] -> [a] -> [a]` și funcția de inversare a unei liste `reverse :: [a] -> [a]`

10. Demonstrați, pornind de la definițiile date la 9, că are loc următoarea relație: `reverse (xs ++ ys) = reverse ys ++ reverse xs`.