

## Závěrečná zkouška z funkcionálního programování IB015

31.5.2012, Jarní semestr, Barnat, 1. termín

1. Vybrat možnost, kde je funkce "foldr1 y (map x [3,4,5])" správně v pointfree. (body 10/0/-2)

**Řešení:** (flip foldr1).(flip map [3,4,5])

---

2. Časová složitost funkce "take x [take x [x]]". (body 10/0/-2)

**Řešení:** Časová složitost je KONSTANTNÍ.

**Poznámka:** take x [take x [x]] :: Int -> [[Int]]

Za x se může dosadit pouze číslo (typ Int): take 2 [take 2 [2]], take 1000 [take 1000 [1000]]

Vždy bude uvnitř pouze jednoprvkový seznam, tudíž take bude brát maximální dostupný počet prvků - jeden.

---

3. Vybrat možnost, kde se funkce vyhodnotí v konečném čase. (body 10/0/-2)

**Řešení:** žádná odpověď není správná

**Poznámka:** každá z odpovědí se zacyklila.

---

4. Nejobecnější typ funkce "\x -> x 3". (14 bodů)

**Řešení:** \x -> x 3 :: Num a => (a -> b) -> b

**Poznámka:** Jde o to, že za "x" můžeme dosadit unární funkci - např. (3+) 3 ~> 6. Ale zákonitě z toho nemusí lézt stejný typ - např. (>0) 3 ~> True.

---

5. Napsat program, který uživatele požádá o zadání názvu souboru, tento textový soubor pak vyrobí v aktuálním adresáři (předpokládáme, že je adresář prázdný) s daným názvem. Obsahem souboru bude jeho název. Možno používat funkce putStr, getLine, writeFile, >>, >>=). (14 bodů)

*writeFile :: FilePath -> String -> IO ()*

**Řešení:** putStr "Zadejte název souboru: " >> getLine >>= \s -> writeFile s s

---

6. Máme stromovou strukturu zadanou následovně: (18 bodů celkem)

*data Tree a = Leaf a | Branch [Tree a] deriving (Show,Eq)*

- a) Vypsat datové konstruktory a jejich aritu. (2 body)

**Řešení:** Leaf - unární, Branch - unární

- b) Napsat hodnotu typu Tree Int za použití minimálně 2 různých Branch. (4 body)

**Řešení:** Branch [Branch [Leaf 4, Leaf 6], Branch []]

- c) Napsat funkci "rozkvet", která každé "Leaf x" změní za "Branch [Leaf x, Leaf x]". (12 bodů)

**Řešení:**

*data Tree a = Leaf a | Branch [Tree a] deriving (Show,Eq)*

rozkvet (Leaf x) = Branch [Leaf x, Leaf x]

rozkvet (Branch y) = Branch (map rozkvet y)