IB015 Zkouška, termín 1 75 minut **Datum: 15. 1. 2019**

Jméno: **Souřadnice:**

0007 *list* *body* 10

Oblast strojově snímaných informací. Své učo a číslo listu vyplňte zleva dle vzoru číslic. Jinak do této oblasti nezasahujte. 0123456789

1. [10 bodů][Haskell] Uveďte nejobecnější typ výrazu `\x y → filter (y <) (concat x)`. Typy elementárních výrazů jsou:

```
(<) :: Ord a ⇒ a → a → Bool
filter :: (a → Bool) → [a] → [a]
concat :: [[a]] → [a]
```

lambda :: a -> b -> c
x y res

X :: [[a]]

lambda :: [[a]] -> a -> [a]
Ord a

concat x :: [a]
(y <) :: Ord a => a -> Bool
filter :: (a -> Bool) -> [a] -> [a]
(y <) (concat x)
filter (y<) (concat) :: Ord a => [a]

(\x y -> filter (y<) (concat x)) :: Ord a => [[a]] -> a -> [a]

2. [−5/0/5 bodů][Haskell] Akce

```
getLine >>= (flip openFile) ReadMode >>= hGetContents >>= putStr
```

je ekvivalentní akci:

f ⇒ f >> g
g

<- getline
f

x <- f
f >>= \x -> g
g

A)

```
do f ← getLine
   h ← ReadMode openFile f
   c ← hGetContents h
   putStr c
```

B)

```
do f ← getLine
   h ← openFile f >> ReadMode
   c ← hGetContents h
   putStr c
```

C)

```
do getLine >>
   \f → openFile f ReadMode >>
   \h → hGetContents h >>
   \c → putStr c
```

D)

```
do f ← getLine >>
   h ← openFile f ReadMode >>
   c ← hGetContents h >>
   putStr c
```

E) žádná z uvedených možností.

Oblast strojově snímaných informací, nezasahujte. ***Druhá strana se neskenuje.***

IB015 Zkouška, termín 1 75 minut **Datum: 15. 1. 2019**

Jméno: **Souřadnice:**

0007 *list* 2 *body* 10

Oblast strojově snímaných informací. Své učo a číslo listu vyplňte zleva dle vzoru číslic. Jinak do této oblasti nezasahujte. 0 1 2 3 4 5 6 7 8 9

3. [10 bodů][Haskell] V programovacím jazyce Haskell zapište intensionálním způsobem seznam všech trojic kladných celých čísel takových, že v každé trojici (x, y, z) platí, že $x = y + z$. Prvky seznamu musí být lexikograficky uspořádány.

x y z
2 = 1 + 1
3 = 1 + 2
3 = 2 + 1
4 = 1 + 3
4 = 2 + 2
4 = 3 + 2

[(x,y,z) | x <- [2..], y <- [1..(x-1)], z <- [(x-y)]]

Oblast strojově snímaných informací, nezasahujte. ***Druhá strana se neskenuje.***

Jméno: | Souřadnice:

0007 list 3 | body 10

Oblast strojově snímaných informací. Své učo a číslo listu vyplňte zleva dle vzoru číslic. Jinak do této oblasti nezasahujte. 0123456789

4. [15 bodů][Haskell] V programovacím jazyce Haskell definujte datový typ `FemaleTree`, který by bylo možné použít pro uložení ženské části rodokmenu prvních manželství, tj. o každé ženě bude uloženo její jméno, jméno jejího prvního manžela a informace o všech jejích dcerách, pokud již nějaké má. Pro ženy bez manžela uložte pouze jejich jméno. Nemanželské potomky a další manželství vůbec neuvažujte.

Dále naprogramujte funkci, která pro hodnoty vámi definovaného typu počítá maximální délku rodinných větví, které končí jménem Arabela (0 pokud se žádná taková větev nevyskytuje).

```
data FemaleTree = Female String String [FemaleTree] | Female String
                  jm. ženy  jm. muže  dcery          jen jm. ženy (bez manžela a dcer)

howManyArabelas :: FemaleTree -> Int
howManyArabelas (Female x) = if x == "Arabela" then 1 else 0
howManyArabelas (Female _ _ daughters) = if m > 0 then m+1
  else 0
  where m = maxA (map (howManyArabelas) daughters)

maxA :: [Int] -> Int
maxA [] -> 0
maxA [x] -> x
maxA (x:y:xs) -> if (x > y) then (maxA (x:xs)) else (maxA (y:xs))
```

IB015 Zkouška, termín 1 | 75 minut | Datum: 15. 1. 2019

Jméno: | Souřadnice:

0007 list 4 body 10

Oblast strojově snímaných informací. Své učo a číslo listu vyplňte zleva dle vzoru číslic. Jinak do této oblasti nezasahujte. 0123456789

5. [10 bodů][Prolog] V programovacím systému Prolog definujte predikát `two/1`, který uspěje právě tehdy, když zadaný seznam hodnot obsahuje alespoň dva prvky, které jsou syntakticky ekvivalentní. Pokud má predikát uspět, musí uspět právě jednou.

```
isEq(X,[XS]):-X==XS,!.
isEq(X,[Y|YS]):-X==Y,!.
isEq(X,[Y|YS]):-X\==Y, isEq(X,YS).

two([X|XS]):-isEq(X,XS),!.
two([X|XS]):-\+isEq(X,XS), two(XS).
```

Oblast strojově snímaných informací, nezasahujte. ***Druhá strana se neskenuje.***