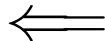


Písomná skúška FP: 2. júna 2023

Táto písomná skúška trvá 90 minút. Počas skúšky je povolené používať laptop s inštalovaným kompilátorom jazyka Haskell `ghc`, resp. `ghci`, `cabal` a/alebo `stack`, `Visual Studio Code` a/alebo `emacs` a/alebo `IntelliJ IDEA`. Iné používateľské aplikácie, najmä komunikačné aplikácie, webové prehliadače a aplikácie pre synchronizáciu súborov, nie sú povolené. Knihy, dokumentácia v papierovej podobe, písané poznámky atď. nie sú povolené.

Túto písomnú skúšku je potrebné vypracovať samostatne, teda bez pomoci niekoho iného a bez komunikácie s niekým iným. Odhalené podvádžanie pri skúške, napr. kopírovanie odpovedí a riešení (aj ich častí), snaha odovzdať cudzie odpovede a riešenia a pod., môže byť penalizované vylúčením študenta z predmetu, nepridelením a/alebo zrušením bodov, prípadne aj disciplinárnym konaním v zmysle Študijného poriadku, ktoré môže viesť k vylúčeniu zo štúdia.

Meno a priezvisko:



1. Prečo nastane chyba pri tomto zápise? Prečo potrebujeme funkciu (\$) ? Ktoré dva vstupné parametre vyžaduje funkcia (\$) ? (5 b.)

```
1 print cos pi
2
3 ($) :: (a -> b) -> a -> b
```

Riešenie:

2. Aký je rozdiel medzi funkciami `seq` a `pseq` ? Prečo ich potrebujeme? (5 b.)

```
1 seq :: a -> b -> b
2 pseq :: a -> b -> b
```

Riešenie:

3. Prečo je prvá možnosť vyhodnotená a vráti hodnotu, a druhá možnosť skončí chybou? (5 b.)

```
1 foldl (\acc x -> x) undefined [1, 2, 3, 4, 5]
2 foldr (\acc x -> x) undefined [1, 2, 3, 4, 5]
```

Riešenie:

4. Naimplementujte funkciu `foo`, ktorá vráti najväčšiu hodnotu zoznamu pomocou `foldl/foldr`. Predpokladajme, že zoznam obsahuje len nezáporné celé čísla. (5 b.)

1 `foo :: Ord a => [a] -> a`

Riešenie:

5. Pomocou `liftA2` naimplementujte funkciu, ktorá vykoná sčítanie prvkov dvoch zoznamov po prvkoch, napr. pre vstup `[1,2]` a `[3,4]` vráti `[4,5,5,6]`. Prečo je riešenie pomocou `liftA2` možné? (5 b.)

Riešenie:

6. Naimplementujte funkciu `foo`, ktorá skombinuje 2 zoznamy rovnakej dĺžky tak, že ak sú obe položky v zozname párne, tak ich sčíta, v opačnom prípade ich prenášobí. Napr. pre vstup `[1,2]` `[3,4]` je výsledná kombinácia `[3,6]`. (5 b.)

Riešenie:

1 `foo :: Integral a => [a] -> [a] -> [a]`

7. Prečo v jazyku Haskell potrebujeme bind operátor $\gg=$? Ako ho využívame? (5 b.)

Riešenie:

8. Aký je rozdiel medzi funkciami `mapM_` a `forM_`? Prečo ich potrebujeme? (5 b.)

```
1 mapM_ :: (Foldable t, Monad m) => (a -> m b) -> t a -> m ()
2 forM_ :: (Foldable t, Monad m) => t a -> (a -> m b) -> m ()
```

Riešenie:

9. Naimplementujte funkciu `foo`, ktorá zjednotí dva zoznamy, pričom duplikátne prvky sú v zjednotenom zozname len raz. Funkciu `foo` implementujte pomocou rekúzie. (10 b.)

Riešenie:

```
1 foo :: (Ord a) => [a] -> [a] -> [a]
```