Programare Functionala 2022
Quiz-uri

PF-C01-Quiz 1

1. Cum se comenteaza o linie in Haskell?
a) -- b) /* */ c) // d) !

2. Ce valoare are x in x = let x = 3 in x * 5?
a) 3 **b) 15** c) 20 d) Eroare

3. Ce valoare are x in x = let x = 3, y = 4 in x * y
a) 3 b) 4 c) 12 **d) eroare**

PF-C01-Quiz 2

1. Ce tip are o functie foo care are doua argumente, primul argument de tip Char, iar al doilea argument de tip Bool, si intoarce un rezultat de tip Bool?
   a) foo : Char -> Bool -> Bool
   b) foo :: Bool -> Char -> Bool
   c) foo :: Char -> Bool -> Bool
   d) nu se poate defini

2. Ce tip are expresia [True, 'a', "FP"]?
   a) (Bool, Char, Char)
   b) eroare
   c) [Bool, Char, [Char]]
   d) [Bool, Char, Char]

3. Ce tip are expresia (True, 'a', "FP")
   a) eroare
   b) (Bool, Char, Char)
   c) (Bool, Char, [Char])
   d) [Bool, Char [Char]]

PF-C02-Quiz 1

1. Ce tip are o functie foo care are doua argumente, o functie de la Char la Bool si, respectiv, un Char, si intoarce un Bool?
a) nu se poate defini
b) foo : (Char -> Bool) -> Char -> Bool
c) foo :: Char -> Bool -> Char -> Bool
d) foo :: (Char -> Bool) -> Char -> Bool

2. Ce valoare are f 3 in f 5 = let x = 3 in x + x?
a) 6
b) 5
c) **exceptie (nu se potriveste niciun caz din definitia lui f)**
d) 10

3. Ce valoare are f 5 in f x = let x = 3 ; y = 4 in x + y?
a) 9
b) 7
c) 5
d) eroare

PF-C02-Quiz 2

1. Cum putem defini lista [3,4,5,6]?
a) 3 : 4 : 5 : 6
b) 3 : 4 : 5 : 6 : []
c) [3 .. 6]
d) 3 : 4 : 5 : [6]

2. Ce obtinem dupa instructiunile?
Prelude> xs = [1,2,3]
Prelude> ys = [11,12]
Prelude> zip xs ys
a) nu se poate aplica functia zip
b) [(1,11),(2,12)]
c) [1,2,3,11,12]
d) [(1,11),(1,12),(2,11),(2,12),(3,11),(3,12)]

3. Ce obtinem dupa instructiunile?
Prelude> let natural = [0..]
Prelude> natural !! 5
a) 6
b) 5
c) [0,1,2,3,4]
d) 4

PF-C03-Quiz 1

1. Fie f x = x + x si g x = x * x. Ce valoare are expresia g . f $ 3?
a) 36
b) 18
c) eroare
d) 6

2. Ce obtinem dupa instructiunea ([1,2,3] ++) [4,5,6]?

a) eroare
b) [1,2,3,4,5,6]
c) [4,5,6,1,2,3]
d) "123456"

3. Ce obtinem dupa instructiunea reverse . take 3 [1 .. 10]?
a) [10,9,8]
b) [1,2,3]
c) eroare
d) [3,2,1]

PF-C03-Quiz 2

1. Ce se obtine dupa instructiunea map (+1) [1,2,3,4]?
a) nu se poate aplica
b) [2,3,4,5]
c) [4,3,2,1]
d) [2,3,4]

2. Ce se obtine dupa instructiunea map (1-) [1,2,3,4]?
a) nu se poate aplica
b) [2,3,4,5]
c) [0,1,2,3]
d) [0,-1,-2,-3]

3. Ce se obtine dupa instructiunea map toUpper "abcd"?
a) nu se poate aplica
b) "dcba"
c) "ABCD"
d) "Abcd"

PF-C03-Quiz 3

1. Ce se obtine dupa instructiunea length . filter (== 'a') "abracadabra"?
a) 5
b) "brcdbr"
c) instructiune invalida
d) "aaaaa"

2. Ce se obtine dupa instructiunea length . filter (== 'a') $ "abracadabra"?
a) 5
b) "brcdbr"
c) instructiune invalida
d) "aaaaa"

3. Ce se intampla dupa instructiunea filter (\x -> (rem x 2) == 0) [1..10]?
a) [2,4,6,8,10]
b) [1,3,5,7,9]

c) 5
d) instructiune invalida

PF-C04-Quiz 1

1. Fie functia foo1 :: (Int,Char,String) -> String. Ce tip are functia curry foo1?
a) nu se poate aplica functia curry peste foo1
b) Int -> Char -> String -> String
c) Int -> (Char -> String) -> String
d) (Int -> Char -> String) -> String

2. Fie functia foo2 :: (Int, (Char,String)) -> String. Ce tip are functia curry foo2?
a) nu se poate aplica functia curry peste foo2
b) Int -> (Char,String) -> String
c) Int -> Char -> String -> String
d) (Int,Char) -> String -> String

3. Fie functia foo3 :: Int -> Char -> String. Ce tip are functia uncurry foo3?
a) nu se poate aplica functia uncurry peste foo3
b) Int -> (Char -> String)
c) functia uncurry nu are niciun efect asupra lui foo3
d) (Int,Char) -> String

PF-C04-Quiz 2

1. Ce se obtine dupa instructiunea foldr (++) ["woot","WOOT","woot"]?
a) "wootWOOTwoot"
b) instructiune invalida
c) ["woot","WOOT","woot"]
d) "woot,WOOT,woot"

2. Ce se obtine dupa urmatoarea instructiune foldr (&&) True [False,True]?
a) instructiune invalida
b) False
c) True
d) [True,False,True]

3. Ce se obtine dupa urmatoarea instructiune foldr (\ x y -> concat ["(",x,"+",y,")"]) "0" ["1","2","3","4","5"]?
a) instructiune invalida
b) "(1+(2+(3+(4+(5+0)))))"
c) "1+2+3+4+5+0"
d) ["(","1","2","3","4","5",")"]

PF-C04-Quiz 3

1. Ce se obtine dupa urmatoarea instructiune foldl (^) 2 [1..3]?
a) 1
b) 64
c) instructiune invalida
d) 8

2. Ce se obtine dupa urmatoarea instructiune foldr (^) 2 [1..3]?
a) 1
b) 64
c) instructiune invalida
d) 8

3. Ce se obtine dupa urmatoarea instructiune foldr (:) [] [1..3]?
a) []
b) instructiune invalida
c) [1,2,3]
d) [3,2,1]

4. Ce se obtine dupa urmatoarea instructiune foldl (flip (:)) [] [1..3]?
a) [1,2,3]
b) instructiune invalida
c) [3,2,1]
d) []

PF-C05-Quiz 1

Fie tipul de date:
```
data Doggies a =
      Husky a
    | Mastiff a
```

1. Ce este Doggies?
a) constructor de tip
b) constructor de date
c) tip de date produs
d) niciunul din raspunsurile de mai sus

2. Ce tip are Mastiff "Scooby Doo"?
a) Doggies
b) [Char]
c) Doggies [Char]
d) Doggies Mastiff

3. Ce tip are Husky (10 :: Integer)?
a) Doggies
b) Doggies Integer

c) Integer
d) Doggies Husky

PF-C06-Quiz 1

1. Clasa Eq
a) include toate tipurile din Haskell
b) coincide cu clasa Ord
c) face testarea egalitatii posibila
d) include doar tipuri numerice

2. Sa presupunem ca clasa de tipuri Ord are operatorul >. Ce tip are >?
a) Ord a => a -> a -> Bool
b) Ord a => Int -> Bool
c) Ord a => a -> Char
d) Ord a => Char -> [Char]

3. Ce puteti sa spuneti despre codul de mai jos?

```
data Mood = Blah
          | Woot deriving Show

settleDown x = if x == Woot
               then Blah
               else x
```

a) codul este corect
b) codul nu este corect deoarece nu exista o instanta a clasei Num pentru tipul Mood
c) codul nu este corect deoarece nu exista o instanta a clasei Ord pentru tipul Mood
d) codul nu este corect deoarece nu exista o instanta a clasei Eq pentru tipul Mood