159272 practice exam questions

Question 1:

a) The expression List((1,2),(3,4)) has type:

i) List[(Int, Int)]
ii) List[Int, Int]
iii) List([Int, Int], [Int, Int])
iv) List(Int, Int)
v) List[Int, Int, Int, Int]

b) The expression List(2,4,6,7).takeWhile(even), where
def even(x:Int): Boolean = (x % 2 == 0), returns:

i) List(2, 4, 6)
ii) List(7)
iii) List(2, 4)
iv) List(2)
v) List[(Int,Int,Int)]

c) Which of the following is true for all finite lists xs

i) xs.reverse.map(f) = xs.map(f).reverse
ii) xs.map(f).map(g) = xs.map(g).map(f)
iii) xs.reverse.reverse = xs.reverse
iv) xs.map(f).map(f) = xs.map(f)
v) xs.reverse = xs

d) Which of the following properties is false:

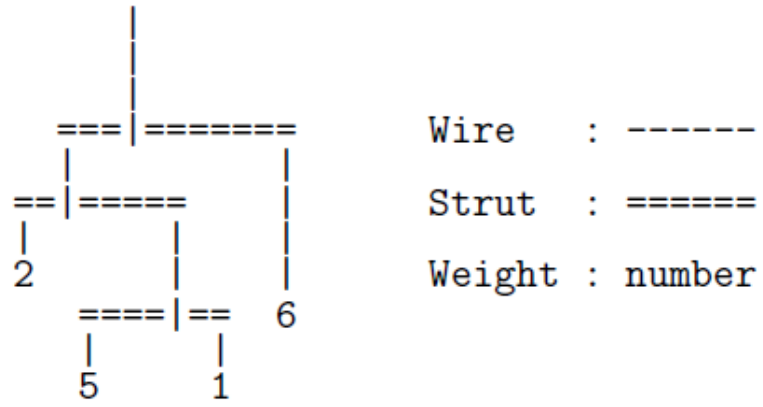
i) x :: List(y,z) = List(x) ::: List(y,z)
ii) List() ::: List(y,z) = List(y,z)
iii) x :: (List(y,z) ::: List(a,b)) = (x :: List(y,z)) ::: List(a,b)
iv) List(x) :: List(y,z) = List(x, y, z)
v) x :: List() = List(x)

e) Evaluating List(1,2,3,4,5).map (_+1).filter(even) gives:
i) List()
ii) List(2, 4)
iii) List(3, 5)
iv) List(2, 4, 6)
v) List(1, 3, 5)

f) Evaluating List(1,2,3,4).map(_+1).foldRight(1)(_*_) gives:
i) 1
ii) 14
iii) 24
iv) 120
v) List(2, 3, 4, 5)

Question 2.

A **mobile** is a hanging ornament made of wires, struts, and weights, such as that illustrated in the following figure:



Mobiles can be modelled using the following algebraic data types:

```
sealed case class Mobile(wire:Int, obj:Object)

sealed abstract class Object
case class Weight(size:Int) extends Object
case class Strut(left:Branch, right:Branch) extends Object

sealed case class Branch(strut:Int, mob:Mobile)
```

That is, a **Mobile** is a pair comprising an integer representing the length of the wire from which it is suspended, and an object attached to the end of that wire. An **Object** is either a **Weight** of a given integer mass, or a **Strut** comprising two branches. Finally, a **Branch** is a pair comprising an integer representing the length of a branch of a strut, and a mobile that hangs from the end of that branch.

a) Draw a labelled picture to illustrate the following mobile:

```
val myMob =
  Mobile(5,Strut(Branch(6,Mobile(2,Weight(2))),Branch(4,Mobile(3,Weight(3)))))
```

b) Assuming that wires and struts are massless, define a function

```
def mass(mob:Mobile): Int
```

that calculates the total mass of a mobile.

c) Define a function

```
def force(b:Branch): Int
```

that calculates the force on a branch, which is given by the product of the length of the branch and the mass of the mobile that hangs from the end.

d) A mobile is balanced if all its struts are horizontal, where a strut is horizontal if the forces on its two branches are equal. Define a function

```
def balanced(mob:Mobile): Bool
```

that decides if a mobile is balanced.

Question 3.

a) Give an example of a higher-order function using Scala.

b) Briefly explain the use and benefits of higher-order functions in functional programming.

c) Give an example of an infinite data structure using Scala.

d) Briefly explain the use and benefits of lazy evaluation in functional programming.

[

+ + + +