

APELLIDOS, NOMBRE:

PROGRAMACIÓN DECLARATIVA CURSO 2018-19
Control Programación Funcional 8-11-2018

- Cada pregunta tiene una única respuesta correcta. Marcad con un aspa la opción elegida.
- **Cada respuesta correcta suma un punto; cada respuesta incorrecta resta medio punto;** las respuestas en blanco ni suman ni restan.

1. Considérense las expresiones: `let x = (++) in x (0:[]) [0]`
`let {x = [0] ++ y ; y = []} in zip x y`
`let x = [0]++(0:[]) in x !! 3`

- ☐ Hay exactamente dos que están mal tipadas.
☐ Hay exactamente una que está mal tipada.
☒ Ninguna está mal tipada.

2. Considérese el operador `infixl 4 ?` y las expresiones: $e_1 = f\ x\ y\ ?\ g\ y\ ?\ x$
 $e_2 = (?)\ (f\ x\ y)\ ((?)\ (g\ y)\ x)$
 $e_3 = (?\ x)\ ((?)\ (g\ y))\ (f\ x\ y)$

- ☒ $e_1 \equiv e_3 \not\equiv e_2$
☐ $e_1 \not\equiv e_2 \not\equiv e_3 \not\equiv e_1$
☐ $e_1 \equiv e_2 \not\equiv e_3$

3. La reducción de la expresión `(\x -> (\x y -> y) x) 2` da como resultado:

- ☐ 2
☒ Una expresión de tipo `a -> a`
☐ Error de tipos.

4. Sea `f` definida por `f x y z = x y z`. El tipo de `f` es:

- ☐ `(a -> b -> c) -> b -> c -> c`
☒ `(a -> b -> c) -> a -> (b -> c)`
☐ `a -> b -> c -> (a -> b -> c)`

5. Sea `f` definida por las siguientes ecuaciones:

`f 0 y z = y`

`f x y z = z`

¿Cuál de las siguientes afirmaciones es cierta?

- ☐ La función no es estricta en ninguno de sus tres argumentos.
☐ La función es estricta en el primer argumento y también en el segundo.
☒ La función es estricta en el primer argumento pero no en el segundo.

6. La evaluación de `foldl (\x y -> x:y) [] [undefined, (True,False)]` da como resultado:

- ☒ Error de tipos.
☐ Error de ejecución.
☐ Una expresión de tipo `[(Bool,Bool)]`.

7. ¿Cuántas de las tres siguientes definiciones de tipos son correctas?

`data T1 a = C1 | C2 (T1 a)`
`data T2 a = C1' Int | C2' (Int,Bool) a`
`data T3 a = C1'' Int Bool | C2'' a`

- ☒ Las tres.
☐ Exáctamente dos de las tres.
☐ Exáctamente una de las tres.
-

8. Considérense las funciones f_1, f_2, f_3 definidas mediante las siguientes ecuaciones:

```
f1 x = uncurry (-) $ (if x > 0 then (x, 1) else undefined)
f2 x = let g = (\(x,y) -> y - x) in (if x > 0 then g (x, 1) else undefined)
f3 x = if x > 0 then ((-) x) 1 else undefined
```

- ☐ $f_1 \equiv f_2$
☐ $f_1 \neq f_3$
☒ Las dos anteriores son falsas.
-

9. Dadas las expresiones: $(\lambda x y \rightarrow y (x: [[1], []]))$
 $(\lambda x \rightarrow (\lambda y \rightarrow y [x, [1], []]))$
 $(\lambda x y \rightarrow y [x, 1])$

- ☐ Las tres tienen el mismo tipo, pero no todas son equivalentes.
☐ Solo dos tienen el mismo tipo, pero no son equivalentes.
☒ Las dos anteriores son falsas.
-

10. La evaluación de `take m (iterate (*i) i) | i <- [1..m]` para un número natural m concreto produce como resultado:

- ☐ Una lista de longitud $m \times m$, cuyos elementos son todas las potencias de 1 a m de cada $i, 1 \leq i \leq m$.
☒ Una lista de longitud m , cada uno de sus elementos representa las potencias i^1, \dots, i^m de un número natural $i, 1 \leq i \leq m$.
☐ Una lista de listas, la lista i -ésima, $1 \leq i \leq m$, tiene como elementos las potencias $1^i, \dots, m^i$.
-

11. La expresión `zipWith (*) (filter p xs) (filter q ys)` se evalúa igual que:

- ☐ `foldr (*) (product $ filter q ys) (filter p xs)`
☐ `concat [[x * y | x <- filter p xs] | y <- filter q ys]`
☒ `foldr f [] (zip (filter p xs) (filter q ys)) where f (x,y) xs = (:) (x * y) xs`
-

12. La evaluación de `foldr (\x y -> x:y) [] [undefined, (True,False)]` da como resultado:

- ☐ Error de tipos.
☒ Error de ejecución.
☐ Una expresión de tipo `[(Bool,Bool)]`.
-

13. ¿A cuál de las expresiones de abajo es equivalente la siguiente lista intensional?

- ```
[x + y | x <- [1..n], p (n-x), y <- [x..m]]
```
- ☒ `concat (map f (filter (\z -> p (n-z)) [1..n])) where f x = map (\y -> x + y) [x..m]`  
☐ `map f [1..n] where f x = filter (p (n-x)) (map (\y -> x + y) [x..m])`  
☐ `(filter (\x -> p (n-x))) $ concat (map f [1..n]) where f x = map (\y -> x + y) [x..m]`
- 

14. Considerando la definición de tipos

```
data T a = C1 | C2 a
 deriving (Show, Eq, Ord)
```

¿Cuál de las siguientes expresiones evalúan a `True`?

- ☐ `compare (C2 'b') C1 == LT`  
☐ `compare (C2 'b') (C2 (C2 'b')) == LT`  
☒ Ninguna de las dos anteriores.
-