

Pregunta 1 (1,5 puntos) Dadas las siguientes definiciones (y declaración de tipo de mifoldr):

```
mifoldr :: (α -> β -> β) -> β -> [α] -> β      -- como foldr pero restringido al tipo [α]
mifoldr = foldr

f p s = filter (p . fst) s

g x [] = []
g x (y:s) = (x==y) : g x s

h x s = mifoldr max x s
```

Da la respuesta del sistema a cada una de las siguientes expresiones y órdenes de tipo (:t)

Expresión/orden

max 5 9

:t max

fst ('u',9)

:t fst

f (>0) [(3,8),(-2,7),(-6,1)]

f (=='a') [('a',8),('p',7),('a',5),('h',3)]

:t f

g 'e' "elefante"

:t g

h 4 [2,3,8,5,2,1]

h "hola" ["adios", "follon"]

:t h

Explica con palabras qué hace la función h

Pregunta 2 (1,5 puntos) Utilizando la función **span**, define una función

$\text{dividirEn} :: \text{Eq } \alpha \Rightarrow \alpha \rightarrow [\alpha] \rightarrow [[\alpha]]$

tal que **dividirEn x s** obtiene la lista de todas las sublistas de **s** que se obtienen al partir **s** en cada aparición de **x**, eliminando ésta.

Ejemplos:

$\text{dividirEn 'a' "parafernalia"} = ["p", "r", "fern", "li", ""]$

$\text{dividirEn 2 [1,2,3,4,2,7]} = [[1], [3,4], [7]]$

$\text{dividirEn 5 [1,3,4]} = [[1,3,4]]$

Ayuda: recordar que $\text{span } p \text{ s} = (\text{takeWhile } p \text{ s}, \text{dropWhile } p \text{ s})$

Pregunta 3 (1,5 puntos) Dados los siguientes tipos

```
type Fecha = (Dia, Mes, Año)
type Dia = Int           -- días de un mes: 1 a 31
type Año = Int
type Mes = Int           -- meses de un año: 1 a 12
type Temperatura = Float
```

3.1 Da el tipo y definición de una función **anteriorA** que decida si una fecha es anterior a otra.

3.2 ¿Cómo comprobarías si el día 3 de abril de 2000 es anterior al 1 de mayo del 2000?

3.3 Define la función **tempMedia** :: [(Fecha, Temperatura)] -> Fecha -> Fecha -> Temperatura que, dada una lista lis de pares (fecha, temperatura) y dos fechas f1 y f2 (con f1 anterior a f2), calcule la media de las temperaturas de la lista lis comprendidas entre las fechas f1 y f2. Si f1 es mayor que f2 debe dar un error con mensaje apropiado.

Pregunta 4 (1,5 puntos) Cambiamos en la pregunta 3, el tipo Fecha por el siguiente tipo algebraico:

```
data MiFecha = F Dia Mes Año  (manteniendo los demás tipos como están).
```

4.1 ¿Cómo harías para que, en el tipo MiFecha, el operador < represente la función **anteriorA**?

4.2 ¿Cómo comprobarías ahora si el día 3 de abril de 2000 es anterior al 1 de mayo del 2000?

4.3 Haz que MiFecha sea instancia de Show, eligiendo una representación adecuada para mostrar fechas. ¿Cómo se mostraría en pantalla el día 3 de abril de 2000?

Pregunta 5 (Optativa) La siguiente función **partesDe** :: [α] -> [[α]] define las partes de un conjunto, P(C), dado el conjunto C como lista:

```
partesDe [] = [[]]
partesDe (x:s) = t ++ map (x:) t  where t = partesDe s
```

5.1 Da el resultado de la expresión **partesDe [1,2,3]**

5.2 El número de elementos de P(C) es 2^n con n = número de elementos de C.

Prueba por inducción que para toda lista finita s se cumple la propiedad:

$$\text{length (partesDe s)} = 2^{\text{length s}}.$$

Nota: Si necesitas resultados auxiliares, los enuncias como lemas (sin probarlos).