

Relación de ejercicios nº 4

1. Explicar como es la evaluación perezosa de las siguientes expresiones:
 - a) `dropWhile (<10) [8..]`
 - b) `drop 5 [8..]`
 - c) `take 2 (map fact [0..])`
 - d) `head (filter (>10) (map fact [0..]))`
 - e) `filter (<100) (map (^2) [1..])`
 - f) `takeWhile (<100) (map (^2) [1..])`
 - g) `takeWhile (/=0) (map (+3) [-5..])`
 - h) `filter p [[1],[1,2],[1,2,3],[1,2,3,4]]`
`where p xs = length xs > 2`
 - i) `filter (not . even . length) xs`
`where xs = [[1],[1,2],[1,2,3],[1,2,3,4]]`
 - j) `foldr (\ xs i -> length xs + i) 0 xs`
`where xs = [[1],[1,2],[1,2,3],[1,2,3,4]]`
 - k) `(fst. head . zip [1,2,3]) [4,5,6]`
2. Utiliza la función `iSort :: Ord a => [a] -> [a]` del Tema 8 para definir una función `minList` que calcule el mínimo de una lista tomando la cabeza de la lista ordenada. ¿Te parece esta una forma ineficiente de resolver el problema? Describe los pasos de evaluación de `minList [8,6,1,7,5]`. ¿Qué ocurre?
3. Dibujar la evaluación de las tres siguientes estructuras cíclicas
 - a) `duda = si where si = "Si." ++ no`
`no = "No." ++ puede`
`puede = "Puede ser." ++ si`
 - b) `facts = 1 : zipWith (*) [1..] facts`
 - c) `fibs = 0 : 1 : [x+y|(x,y)<- zip fibs (tail fibs)]`
4. Definir con una estructura cíclica la lista infinita de los números pares
5. Definir usando estructuras cíclicas la siguiente tabla infinita de potencias

```
tabla = [ [1,2,4,8,...
          [1,3,9,27,...
          [1,4,16,64,...          ...
```

y una función que, dados dos enteros n y m , obtenga el fragmento de las m primeras filas y n primeras columnas de dicha tabla.
6. Definir usando estructuras cíclicas una función
`aplicar f e [x1,x2,... = [e, f x1 e, f x2 (f x1 e),...`
Utilizar dicha función para obtener la lista infinita creciente de todos los números perfectos. Además definir una función que, dado un entero n , calcule el menor número perfecto que supera a n .
Indicación: Un número es perfecto si es de la forma $1+2+3+\dots+n$, para algún entero positivo n .