UNIVERSIDAD DE MÁLAGA
E.T.S.I. INFORMÁTICA

Programación Declarativa

(3º de Ingeniería Técnica en Informática)
Examen de Septiembre de 2007

Nombre:
Especialidad: **Grupo:**

Ejercicio 1

a. (*1 pto.*) ¿Cuál es el tipo polimórfico de `fun`?

```
fun f g x = f g x
```

b. (*1 pto.*) ¿Cuál es el tipo polimórfico de `fun'`?

```
fun' f g x = f . (g x)
```

c. (*1 pto.*) ¿Cuál es el tipo polimórfico de `fun''`?

```
fun'' f g x = (f g) . x
```

Ejercicio 2

2.1 (*1 pto.*) Definir una función `agrupar`, dando su tipo, que tenga como argumento una lista con elementos posiblemente repetidos y devuelva otra lista ordenada de pares en la que aparezca cada elemento de la primera lista con el número de veces que se encuentra en ella.

2.2 (*1 pto.*) Expresar la función anterior como un caso particular de `foldr`.

2.3 (*1 pto.*) Dadas las siguientes definiciones de tipos:

```
type Codigo = Integer
type Unidades = Integer
type Nombre = String
type PrecioUnitario = Double
type Producto =(Codigo,Nombre,PrecioUnitario)
type BD = [Producto]
type Compra = [Codigo]
type Albaran = [(Unidades,Nombre,PrecioUnitario)]
```

definir una función `mkAlbaran :: BD ->Compra ->Albaran` que, dada una compra en la que pueden aparecer códigos de productos repetidos en cualquier orden, y una base de datos con las características de cada producto produzca el correspondiente albarán.

2.4 (*1 pto.*) Expresar esta función con ayuda de `foldr`

Ejercicio 3

Sea la siguiente definición de tipo

```
data Problema a = Cal a | Sub [Problema a] deriving Show
```

para representar los puntos correspondientes a los distintos apartados de problemas de un examen. Por ejemplo:

```
ej = Sub [ Sub [Cal 0.75, Cal 1.25, Cal 1]
         , Cal 3
         , Sub [Cal 2.5, Cal 1.5]
         ]
```

representa un examen con tres problemas, donde el primero consta de tres apartados (valorados en 0.75, 1.25 y 1 puntos respectivamente), el segundo problema vale 3 puntos y el tercero consta de dos apartados (de 2.5 y 1.5 puntos).

a. (*1 pto.*) Define la siguiente función de plegado para el tipo `Problema a`:

```
foldProblema :: ([b] -> b) -> (a -> b) -> Problema a -> b
```

b. (*1 pto.*) Usando la función `foldProblema`, define una función `notaTotal` que calcule la puntuación total del un examen. Por ejemplo,

```
notaTotal ej ==> 10.0
```

Indica además el tipo sobrecargado de la función definida.

c. (*1 pto.*) Usando la función `foldProblema`, define una función `maximaNota` que devuelva la nota con mayor valoración de un examen. Por ejemplo,

```
maximaNota ej ==> 3.0
```

Indica además el tipo sobrecargado de la función definida.