

20 de diciembre de 2011

Apellidos: ..... Nombre: .....

**INSTRUCCIONES**

- Resuelve el examen en un archivo que tenga por nombre `laboratorio-DNI-NOMBRE.hs`, sustituyendo DNI por tu número de dni o pasaporte y NOMBRE por tus apellidos y nombre (separados por guiones).
- Escribe también lo siguiente en las primeras líneas de ese archivo: dni, apellidos y nombre, nombre del ordenador desde el que estás realizando el examen.

**NOTA:** en este examen es **obligatorio** especificar, de la forma más general posible, el tipo de las *funciones solicitada*. Para las demás funciones auxiliares que se definan no es necesario.

### Ejercicio 1

Se dispone de dos jarras, sin marcas de medición, pero de las que se sabe sus capacidades máximas. Hay un grifo que permite llenar las jarras completamente de agua. También se puede verter el contenido de una jarra en la otra, bien quedando vacía la jarra desde la que se vierte el agua, bien quedando llena la jarra en la que se vierte el agua. El problema consiste en obtener una cantidad prefijada de agua en una de las jarras.

Definir el tipo de datos `Jarra` como un sinónimo de una tupla de dos números de tipo **Int** (representando el contenido y la capacidad máxima de una jarra) y el tipo de datos `Vertido` como un sinónimo de una cadena ("1 a 2" o "2 a 1", representando el vertido del contenido de una jarra en la otra).

Definir un nuevo tipo de datos `ProblemaDeLasJarras` que tenga un único constructor con dos argumentos que sean del tipo `Jarra`. Este tipo de datos representa un posible estado del problema. Siempre asumiremos que al construir un valor de este tipo de datos los argumentos proporcionados son correctos.

Definir la función `verter` que reciba un estado del `ProblemaDeLasJarras` y un `Vertido` a realizar y devuelva el estado del `ProblemaDeLasJarras` resultante de realizar el vertido indicado. Siempre asumiremos que todos los argumentos recibidos son correctos. Para simplificar los cálculos, también admitiremos como definición válida una en la que la jarra que vierte su contenido siempre quede vacía (es decir, el contenido sobrante del vertido se desecha).

## Ejercicio 2

Consideremos la siguiente definición de un nuevo tipo de dato que representa de manera recursiva los polinomios con coeficientes enteros:

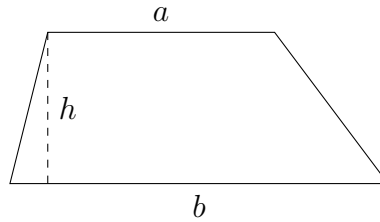
```
type Termino a = (a, a)
data Polinomio a = PolCero | Pol (Termino a) (Polinomio a)
deriving Show
```

Es decir, un polinomio con coeficientes enteros es el polinomio cero ( $0x^0$ ) o un polinomio obtenido añadiendo un nuevo término con un cierto coeficiente entero y un cierto grado a un polinomio ya existente. Siempre asumiremos que no añadimos un término de grado igual a uno del polinomio ya existente, con la única excepción de que el término sea de grado 0, que se podrá añadir si el único término de grado 0 del polinomio ya existente es  $0x^0$ . Por otra parte, los términos no tienen por qué añadirse en orden creciente de grado.

Definir la función `valor` que reciba un `Polinomio` y un número real y devuelva el valor que toma el polinomio en ese número (es decir, el valor que se obtendría al sustituir en el polinomio la variable  $x$  por el número real especificado). Siempre asumiremos que el polinomio recibido está construido de manera correcta. Recuérdese la existencia de la función **`fromIntegral`** de transformación de tipos numéricos.

### Ejercicio 3

Un trapecio es un cuadrilátero que tiene dos lados paralelos y otros dos que no lo son. Los lados paralelos  $a$  y  $b$  se llaman bases del trapecio y la distancia  $h$  entre ellos se llama altura.



Recordando que el área de un trapecio se calcula mediante la fórmula  $\frac{a+b}{2}h$ , definir una función que pida desde el teclado las bases y la altura de un trapecio y escriba en pantalla su área.