

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

Las Torres de Hanoi es un juego matemático. Consiste en tres varillas verticales y un número indeterminado de discos que determinarán la complejidad de la solución. No hay dos discos iguales, están colocados de mayor a menor en la primera varilla ascendentemente, y no se puede colocar ningún disco mayor sobre uno menor a él en ningún momento. El juego consiste en pasar todos los discos a la tercera varilla colocados de mayor a menor ascendentemente.



En este ejercicio representamos las varillas del juego con las cadenas "I", "C" y "D" y los tamaños de los  $n$  discos del juego con los números enteros de 1 a  $n$ .

Definir el tipo de datos `PilaDeDiscos` como un sinónimo de una lista de números de tipo **Int** y el tipo de datos `Varilla` como un sinónimo de cadena.

Definir un nuevo tipo de datos `TorreDeHanoi` que tenga un único constructor con tres argumentos que sean del tipo `PilaDeDiscos`. Siempre asumiremos que al construir un valor de este tipo de datos las pilas de discos proporcionadas son correctas.

Definir la función `moverDisco` que reciba una `TorreDeHanoi` y dos `Varillas` y devuelva la `TorreDeHanoi` resultante de mover un disco de la primera a la segunda `Varilla` proporcionadas. Siempre asumiremos que todos los argumentos recibidos son correctos y que el movimiento se puede realizar.

## Ejercicio 2

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

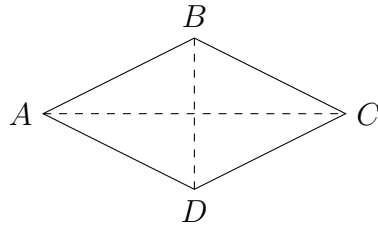
```
type Termino a = (a, a)
data Polinomio a = PolCero | Pol (Termino a) (Polinomio a)
deriving Show
```

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 `grado` que reciba un `Polinomio` y devuelva el grado de ese polinomio. Siempre asumiremos que el polinomio recibido está construido de manera correcta.

### Ejercicio 3

Un rombo es un polígono de cuatro lados paralelos dos a dos y todos de igual longitud.



Recordando que el área de un rombo se puede calcular como el semiproducto de sus diagonales,  $\frac{\overline{AC} \cdot \overline{BD}}{2}$ , definir una función que pida desde el teclado las diagonales de un rombo y escriba en pantalla su área.