

Apellidos,
Nombre

DNI

La puntuación total del examen es de 10 puntos. Para aprobar será necesario un mínimo de 5 puntos.
Para todas las funciones que definas, indica además su tipo.

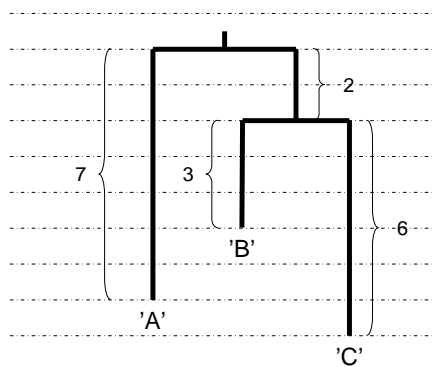
1) Considera el siguiente tipo para representar árboles binarios con valores tan solo en sus hojas, para los que además se especifica la longitud de cada una de las ramas:

```

type Long = Integer
data ÁrbolB a = HojaB a | NodoB Long (ÁrbolB a) Long (ÁrbolB a)
                deriving Show
    
```

Así, una hoja del árbol que almacene un valor *v* se representa con `HojaB v`, mientras que un nodo interior se representa con `NodoB li ai ld ad`, donde *li* es un entero que indica la longitud de la rama izquierda, *ai* es el subárbol izquierdo, *ld* la longitud de la rama derecha y *ad* el subárbol derecho.

Por ejemplo, el siguiente árbol:



sería representado como:

```

ejemplo :: ÁrbolB Char
ejemplo = NodoB 7 ai 2 ad
  where
    ai = HojaB 'A'
    ad = NodoB 3 adi 6 add
    adi = HojaB 'B'
    add = HojaB 'C'
    
```

- a) (1.5 puntos) Sea la profundidad de un valor almacenado en un árbol la suma de las longitudes de las ramas por las que hay que pasar para llegar desde la raíz del árbol hasta dicho valor. Escribe una función `másProfundoB` que, dado un árbol, devuelva el valor almacenado en él a máxima profundidad. Por ejemplo, `másProfundoB ejemplo ==> 'C'` ya que dicho nodo está a profundidad 8.

AYUDA: define primero otra función que devuelva en una tupla tanto el valor a máxima profundidad como dicha profundidad.

- b) (0.5 puntos) ¿Qué devolvería la función `másProfundoB` que has definido en el apartado anterior si hay varios valores con profundidad máxima?

Sea el siguiente tipo para representar árboles generales con las mismas características que los definidos anteriormente:

```
data Árbol a = Hoja a | Nodo [ (Long, ÁrbolB a) ] deriving Show
```

donde cada tupla en la lista es un subárbol hijo junto con la longitud de la rama que lleva a él.

- c) (1.5 *puntos*) Define una función `másProfundos :: Árbol a -> [a]` que, dado un árbol general, devuelva una lista con todos los valores que se encuentran a máxima profundidad.

2) NOTA: para este problema, no se aceptará como solución convertir los enteros al tipo `Integer` y operar éstos, es decir, las funciones definidas deben operar directamente valores de tipo `Entero`.

Sea el siguiente tipo para representar números enteros de modo similar a la definición del tipo `Nat` visto en clase:

```
data Entero = Cero | Suc Entero | Pred Entero deriving Show
```

de modo que `Cero` representa el número 0, `Suc n` representa el sucesor de `n` y `Pred n` el predecesor de `n`.

- a) (1.5 *puntos*) Define un operador `(<+>) :: Entero -> Entero -> Entero` que calcule la suma de dos enteros.
- b) (2 *puntos*) Observa que, mediante este tipo, un mismo valor puede representarse de varias formas. Así, el 2 puede representarse como `Suc (Suc Cero)` o `Suc (Pred (Suc (Suc Cero)))`, por ejemplo. Escribe una función `simplificar`, que dado un valor de tipo `Entero` devuelva la representación más simple del mismo. Por ejemplo,

```
simplificar (Pred (Pred (Suc (Suc Cero)))) ==> Cero
```

3)

- a) (1.5 *puntos*) Define una función `longSuma`, que dada una lista de valores reales, devuelva una tupla con dos componentes. La primera será el número de elementos de la lista y la segunda la suma de todos los valores de la lista. **Define esta función usando foldr**. Por ejemplo:

```
longSuma [1.0, 2.5, 7.0] ==> (3,10.5)
```

- b) (0.5 *puntos*) Usando la función anterior, define una función que devuelva la media de los elementos de una lista.
- c) (1 *punto*) La mediana de un conjunto de valores es aquél valor tal que el 50% de los valores del conjunto se encuentra por debajo de él y el otro 50% se encuentra por encima. Si el número de valores es impar, la mediana coincide con alguno de los valores, por ejemplo:

```
mediana [3, 20, 1, 10, 50] ==> 10
```

ya que el 1 y el 3 son menores a 10, y además el 20 y el 50 son mayores a 10. Si el número de valores es par, la mediana es la media de los dos valores centrales. Por ejemplo:

```
mediana [3, 20, 1, 10, 50, 15] ==> 12.5
```

ya que los elementos centrales son 10 y 15, y la media de éstos es 12.5. Escribe una función que, dada una lista de valores reales, devuelva la mediana.

AYUDA: prueba a ordenar los elementos del conjunto.