

UNIVERSIDAD
COMPLUTENSE
DE MADRID



Ejercicios propuestos - Colecciones

Colecciones

1. Escribir en Scala el código equivalente a:
`for (int i=10; i>=0; i--) println(i)`

tip: existen generadores de colecciones:

```
val l = 0 to 10
```

2. Definir una función cuya firma es: `countDown(n: Int): Unit`. Esta función imprimirá los números de `n` a 0
3. Escribir el código que asigne a la variable 'a' una colección (da igual si se define para, `Seq`, `List`, `Array`) de `n` enteros aleatorios entre 0 (incluido) y `n` (excluido)

tip: usar el singleton Object: `scala.util.Random` que ofrece funciones de utilidades como `nextInt`, que acepta un argumento de tipo `Int` que indica el límite superior del rango de números de los cuales generará uno aleatorio.

4. Dado una colección de enteros, se pide generar una nueva colección que contenga todos los números positivos de la colección original en el orden de la primera colección seguidos por los ceros y negativos, todos en su orden original
5. Definir una función que calcule la media de un `Array[Double]`
6. Definir una función que reciba un argumento de tipo `Array[Int]` y devuelva un `Array[Int]` sin duplicados.
7. Definir una función que reciba un argumento de tipo `Seq[Int]` y devuelva otra secuencia sin ceros.
8. Definir una función que reciba un argumento de tipo `Map[String, Int]` y produzca un `Map[String, Int]` manteniendo las mismas claves, pero con los valores incrementados en 100.

Ejemplo de entrada: `Map("a"->10, "b"->20, "c"->100, "d"->200)`

Ejemplo de salida: `Map("a"->11, "b"->22, "c"->110, "d"->220)`

9. Definir una función que reciba una colección: `minmax(values: Array[Int])` que devuelva un par (tupla) con el menor y mayor valor del array.
10. Definir una función que reciba como argumento un `String` y produzca un mapa con los índices de todos los caracteres. Por ejemplo: Si recibe "albacete", se produzca un mapa: `Map("a" -> [0, 3], "l" -> [1], "b" -> [2] ...)`