

PROGRAMACIÓN LÓGICA Y FUNCIONAL  
PRACTICA 3 "LISTAS"

Escriba funciones en Haskell que resuelvan los siguientes problemas (puede escribir funciones auxiliares si lo requiere):

1. Función **ordenada** que recibe una lista y regresa True si está ordenada ascendentemente o False si no. Ejemplos:

```
*Main> ordenada [3,4,5,10,20]
True
*Main> ordenada [3,4,5,4,20]
False
```

2. El algoritmo "**Merge Sort**" ordena una lista realizando los siguiente pasos:
  - a. Divide la lista en dos partes (la mitad de elementos en cada una)
  - b. Ordena recursivamente (utilizando Merge Sort) cada lista
  - c. Mezcla ambas listas ordenadas

Escribir la funcion **mergeSort** que recibe una lista y regresa la lista ordenada ascendentemente.

Ejemplos:

```
*Main> mergeSort [5,9,1,4,10,9,2,8,13,7]
[1,2,4,5,7,8,9,9,10,13]
*Main> mergeSort [5]
[5]
*Main> mergeSort []
[]
```

3. El algoritmo "**Quick Sort**" ordena una lista realizando los siguientes pasos:
  - a. Elige un elemento "pivote" (puede ser la cabeza de la lista)
  - b. Construye dos listas, una con todos los elementos menores o iguales al pivote y otra con los elementos mayores al pivote
  - c. Ordena recursivamente ambas listas utilizando "quick sort"
  - d. Concatena ambas listas, poniendo en medio de la concatenación al elemento pivote.

Escribir la funcion **quickSort** que recibe una lista y regresa la lista ordenada ascendentemente.

Ejemplos:

```
*Main> quickSort [5,9,1,4,10,9,2,8,13,7]
[1,2,4,5,7,8,9,9,10,13]
*Main> quickSort [4]
[4]
*Main> quickSort []
[]
```