

Apellidos:**Nombre:****Grupo 2A**

Ejercicio 1 Define la función

```
iguales :: Eq a => [a] -> [a]
```

tal que (iguales xs) devuelve el mayor prefijo de xs formado por elementos iguales. Por ejemplo:

```
iguales [1,1,1,2,3,1] ==> [1,1,1]
```

```
iguales "aabbcc" ==> "aa"
```

```
iguales [True, False, True, True] ==> [True]
```

Usaremos el siguiente tipo de dato para representar un grafo no dirigido.

```
data Grafo a = G [a] [(a,a)] deriving Show
```

El grafo de vértices xs y de aristas ys vendrá dado por (G xs ys) Por ejemplo, el grafo que tiene por vértices ['a','b','c','d','e'] y por aristas [(('a','b'), ('b','c'), ('b','d'), ('c','d'))], vendrá representado por:

```
g1 = G ['a','b','c','d','e']  
      [(('a','b'), ('b','c'), ('b','d'), ('c','d'))]
```

Ejercicio 2 Define la función aislados :: Eq a => Grafo a -> [a]

tal que (aislados g) devuelve la lista de los vértices aislados de g. Por ejemplo,

```
aislados g1 ==> e
```

```
aislados (G [1,2,3,4] []) ==> [1,2,3,4]
```

```
aislados (G [1,2,3] [(1,2), (1,3), (2,3)]) ==> []
```

Una coloración de un grafo asocia a cada vértice un color de manera que vértices adyacentes no están asociados al mismo color. Representaremos los colores por números enteros y las asociaciones de colores a vértices por listas de pares vértice-color.

```
type Color = Int
```

```
type Asociacion a = [(a,Color)]
```

Ejercicio 3 Define el predicado

`esAsociacion :: Eq a => Grafo a -> Asociacion a -> Bool`
tal que `(esAsociacion g cs)` decide si `cs` es o no una coloración de `g`.
Por ejemplo, para el grafo del ejercicio anterior, se tiene que de las siguientes asociaciones de colores a vértices,

```
c1,c2 :: Asociacion Char
c1 = [('a',1),('b',2),('c',1),('d',3)]
c2 = [('a',1),('b',2),('c',1),('d',2)]
esColoracion g1 c1 ==> True
esColoracion g1 c2 ==> False
```