Tema A

Ejercicio 1: Definir el tipo EmpresaTelefono cuyos constructores serán Claro, Personal, Movistar y Tuenti. Defini un tipo Frase como sinónimo de String y defini la función

```
fraseEmpresa :: EmpresaTelefono -> Frase
```

que a cada empresa le asocia su frase ("Claro, La red mas poderosa", "Personal, es como vos", "Movistar, Compartida la vida es mas... ", "Tuenti es la mas economica"). **Usar Pattern Matching**.

Dé un ejemplo de ejecución.

Ejercicio 2: Una persona puede tener cuenta en varias empresas, vamos a llevar el registro de las empresas de teléfonos por persona. Defina el tipo NombrePersona como sinónimo de String. Defina el tipo recursivo MisEmpresas que tiene 2 constructores:

- AgregaEmpresa: guarda que EmpresaTelefono estoy agregando, un NombrePersona y un valor de tipo MisEmpresas a la cual se le agrega la EmpresaTelefono con el NombrePersona.
- Ninguna: sin parámetros y es equivalente a la lista vacía.

Luego programar la función

```
tengoEmpresa :: MisEmpresas -> EmpresaTelefono -> NombrePersona -> Bool
```

que dado un valor del tipo MisEmpresas, una EmpresaTelefono y un NombrePersona, devuelve True si la EmpresaTelefono y el NombrePersona están en MisEmpresas, False en caso contrario.

Dé un ejemplo de ejecución que incluya su **propio nombre** como algún valor de tipo NombrePersona.

Ejercicio 3: Defina el tipo NroTel como sinónimo de Int, usando la definición de ListaAsoc del Proyecto 2 programar la función

```
agregaLA :: ListaAsoc EmpresaTelefono NroTel -> EmpresaTelefono ->
NroTel -> ListaAsoc EmpresaTelefono NroTel
```

que devuelve la lista de asociaciones a la cual le agrego la asociación NroTel con la EmpresaTelefono.

Dé un ejemplo de ejecución.

Ejercicio 4*: Usando la definición de árboles binarios del Proyecto 2, Ejercicio 7* defina la siguiente función

```
aBusca :: Eq a => Arbol a -> a -> Bool
```

que dado un árbol y un valor de tipo a devuelve True si el valor está en el árbol y False en caso contrario.

Dé un ejemplo de ejecución.