



Integradores

Liga de Fútbol Femenino

Una agrupación de ligas de fútbol femenino quiere agregar funcionalidad a su sistema informático que al momento cuenta con los siguientes tipos:

```
type LigaFemenina is record { /*  
PROPOSITO: Modelar una liga femenina de fútbol.  
INV. REP.:  
* La lista **equipos** no tiene equipos con el mismo nombre.  
* La lista **sponsors** no tiene sponsors con el mismo nombre.  
* Todo equipo de la lista **equipos** es apoyado por uno o más  
    sponsors de la lista **sponsors**.  
* Cada nombre de sponsor que apoya a un equipo de la lista  
    **equipos** corresponde al nombre de un sponsor de la lista  
    **sponsors**.  
*/  
field sponsors // [sponsor]  
field equipos // [Equipo]  
}
```

```
type Rubro is  
variant { /*  
PROPOSITO: Modelar  
los rubros de los  
sponsors de la  
liga.  
*/  
case Cripto {}  
case Seguros {}  
case Bebidas {}  
case Salud {}  
case Automotriz {}  
}
```

```
type Equipo is record { /*  
PROPOSITO: Modelar un equipo de la liga.  
INV.REP.:  
* **nombre** no es un string vacío.  
* La lista **nombreDeSponsors** no tiene  
    repetidos.  
*/  
field nombre // String  
field nombresDeSponsors // [String]  
}
```

```
type Sponsor is record { /*  
PROPOSITO: Modelar un sponsor.  
INV. REP.:  
* **nombre** no es un string vacío.  
* **montoAportado** es mayor a cero.  
*/  
field nombre // String  
field rubro // Rubro  
field montoAportado // Número  
}
```

Usando este modelo como base, considere las siguientes situaciones:

Punto 1)

Cuando un sponsor retira su apoyo a una liga de fútbol femenino, debe removérse de la liga toda referencia a dicho sponsor.

Por lo cual se pide que implemente la función **liga_SinSponsorLlamado_**, que dada una liga y el nombre de un sponsor existente en la liga, describe la liga que resulta de quitar ese sponsor de la lista de sponsors de la liga, así como de todos aquellos equipos que pudieran tenerlo como sponsor.

Pista: Recuerde qué funciones de biblioteca tiene disponibles para usar al procesar listas.

Punto 2)

En general los sponsors prefieren evitar publicitar en ligas que tengan mucha presencia de la competencia. Por eso, a la hora de decidir a qué ligas apoyar buscan aquella que tenga el menor número de equipos con sponsors de su mismo rubro. Se pide entonces que implemente la función **mejorLigaEntre_ParaSponsor_** que dados una lista de ligas femeninas y un sponsor, describa a aquella liga de entre las dadas que tiene el menor número de equipos con sponsors del rubro del sponsor dado.

Se sugiere que su estrategia incluya la implementación de la subtarea **sponsorsDeEntre_NombradosEn_** que dada una lista de sponsors y una lista de strings que representan nombres de sponsors, describe una nueva lista conformada por aquellos sponsors de la primera lista cuyos nombres aparecen en la segunda lista. **Aclaración:** esta subtarea es sólo una sugerencia y, en caso de optar por usarla DEBE implementarla ya que no se propone como primitiva.



Pista: Recuerde qué funciones de biblioteca tiene disponibles para usar al procesar listas.

Punto 3)

Considere la siguiente implementación de la función `nuevaMarcaDe_ConMontoAInvertir_DeNombre_`.

```
function nuevaMarcaDe_ConInversionDe_DeNombre_(sponsorExistente, monto, nombreDeSponsor) {
    /*
        PROPÓSITO: Describir un nuevo sponsor que comparte rubro con el
                  sponsor existente dado, que aporta el monto dado y tiene por
                  nombre
                  *nombreDeSponsor*.

        PARÁMETROS:
        * sponsorExistente : Sponsor - El sponsor matriz.
        * monto : Número - El monto que va invertir.
        * nombreDeSponsor : String - El nombre del nuevo sponsor.

        PRECONDICIONES:
        * El monto dado debe ser mayor a 0.
        * El nombre dado no es un string vacío.

        TIPO:
        Sponsor
    */
    return ( Sponsor( sponsorExistente | montoAportado <- monto,
                      nombre <- nombreDeSponsor,
                      rubro <-rubro(sponsorExistente)
                    ))
}
```

Se pide que determine si la solución propuesta es adecuada o no, y **JUSTIFIQUE** su respuesta (la justificación no puede tener más de 3 renglones).