

# Enunciado:

Una empresa China de desarrollo tecnológico en razón a un bloqueo comercial que ha sufrido, esto ha generado grandes pérdidas de patrimonio, capital y ventas. Es por ello, que ha decidido cambiar su razón social,

y ahora se llamará "Juaguey". Esta empresa necesita un nuevo sistema para poder organizar sus sucursales y ventas, para ello:

1. La empresa tiene el **nombre** de su **gerente** y **dos referencias**, La primera hacia los **productos** que vende la compañía y la segunda hacia las **sucursales** que tiene mundialmente.
2. Los **productos** son manejados a través de una lista simplemente enlazada circular. Cada producto tiene **código**, **nombre**, **valor** en dólares y **tipo**.
3. Cada **sucursal** tiene número **id**, **país**, **nombre del encargado** de la **sucursal** y un arreglo compacto de tamaño `maxVentas` que contiene referencias a las ventas, que ha realizado la sucursal. Las **sucursales** se manejan a través de una lista doblemente enlazada.
4. Cada venta tiene **código único**, **fecha** y un arreglo de tamaño `maxProductos` de **referencias a los productos vendidos** en esa boleta.

Se pide **implementar** la siguiente función:

```
struct Sucursal * quitarSucursalMenosVentas(struct Juagey *HW)
```

Esta función recibe por parámetro una referencia a la empresa Juaguey y debe quitar la sucursal que tiene el **monto menor** acumulado en ventas. Para ello, deberá determinar cuál es la **sucursal** que tiene el **monto** acumulado en **ventas menor** en todo el sistema, quitarla de la lista y luego retornar un puntero a la sucursal quitada. en caso de empate quita la primera. En caso de fracaso retorna NULL. No puede asumir que todas las sucursales tienen montos acumulados  $\geq 0$ . No puede usar arreglos, ni listas auxiliares.