1. En Coria del Río, en las marismas del Guadalquivir hay un sistema trasbordador que, de usarlo, permite ahorrar 30 km en automóvil con sólo 300 m de trasbordaje por el río. En este sistema, hay dos barcazas que admiten 4 automóviles cada una y varias personas. Dado los bajos precios que son necesarios para que sea un servicio bastante usado, los propietarios nos piden implementar un sistema que permite maximizar el rendimiento minimizando los tiempos de espera. Para ello nos piden un sistema con las siguientes condiciones: 3

   - Una barcaza no sale hasta que no esté llena de automóviles.
   - Una barcaza no admite automóviles si la otra barcaza tiene algún automóvil y aún no ha salido.
   - Las barcazas pueden estar o no al mismo tiempo en el embarcadero.
   - Sólo hace falta implementar uno de los embarcaderos porque se supone que el otro es idéntico en comportamiento.
   - Los automóviles suben en la barcaza que tenga abiertas sus puertas y si las tienen abiertas las dos barcazas los automóviles suben en una de ellas aleatoriamente (son introducidos por personal de la empresa).

   Implementa los procesos automovil() y barcaza() usando semáforos. Para ello puedes usar las funciones up(sem), down(sem), up(sem,cant) y down(sem,cant).

Solución:

```
#define MAXAUTO 4

semaforo esperando=0, auto=0, embarca=1, puertas=1;
int cuentaPers=0, cuentaAuto=0;

automovil()
{
    llegaAuto();
    down(auto);
    down(embarca);
    embarcando();
    ++cuentaAuto;
    if(cuentaAuto == 4)
    {
        cuentaAuto = 0;
        up(esperando);
    }
    up(embarca);
}

barcaza()
{
    llegaBarcaza();
    down(puertas);
    abrePuerta();
    up(auto,4);
    down(esperando);
    up(puertas);
    saleBarcaza();
}
```

2. En cierto instante un sistema operativo tiene 7 procesos con un conjunto de recursos asignados tal y como se muestra en las tablas adjuntas. (E=recursos totales, A=recursos disponibles, C=matriz de asignados, R=matriz de solicitudes): $\boxed{3}$

Determina paso a paso si el sistema es seguro o inseguro.

$$E=\begin{Bmatrix}5 & 4 & 3 & 6 & 5\end{Bmatrix} \quad A=\begin{Bmatrix}3 & 1 & 2 & 2 & 1\end{Bmatrix} \quad C=\begin{Bmatrix}0 & 0 & 1 & 2 & 0\\0 & 1 & 0 & 0 & 1\\0 & 1 & 0 & 0 & 0\\0 & 0 & 0 & 1 & 1\\0 & 1 & 0 & 0 & 1\\0 & 0 & 0 & 1 & 1\\2 & 0 & 0 & 0 & 0\end{Bmatrix} \quad R=\begin{Bmatrix}3 & 3 & 2 & 2 & 3\\5 & 2 & 3 & 2 & 2\\5 & 3 & 0 & 5 & 2\\5 & 3 & 0 & 5 & 2\\5 & 1 & 1 & 2 & 1\\5 & 1 & 2 & 1 & 1\\3 & 1 & 2 & 1 & 1\end{Bmatrix}$$

Solución:

$$A=\begin{Bmatrix}5 & 1 & 2 & 2 & 1\end{Bmatrix}$$

$$C=\begin{Bmatrix}0 & 0 & 1 & 2 & 0\\0 & 1 & 0 & 0 & 1\\0 & 1 & 0 & 0 & 0\\0 & 0 & 0 & 1 & 1\\0 & 1 & 0 & 0 & 1\\0 & 0 & 0 & 1 & 1\\- & - & - & - & -\end{Bmatrix} \quad R=\begin{Bmatrix}3 & 3 & 2 & 2 & 3\\5 & 2 & 3 & 2 & 2\\5 & 3 & 0 & 5 & 2\\5 & 3 & 0 & 5 & 2\\5 & 1 & 1 & 2 & 1\\5 & 1 & 2 & 1 & 1\\- & - & - & - & -\end{Bmatrix}$$

Comentario:
Se podría atender a P_7

$$A=\begin{Bmatrix}5 & 1 & 2 & 3 & 2\end{Bmatrix}$$

$$C=\begin{Bmatrix}0 & 0 & 1 & 2 & 0\\0 & 1 & 0 & 0 & 1\\0 & 1 & 0 & 0 & 0\\0 & 0 & 0 & 1 & 1\\0 & 1 & 0 & 0 & 1\\- & - & - & - & -\\- & - & - & - & -\end{Bmatrix} \quad R=\begin{Bmatrix}3 & 3 & 2 & 2 & 3\\5 & 2 & 3 & 2 & 2\\5 & 3 & 0 & 5 & 2\\5 & 3 & 0 & 5 & 2\\5 & 1 & 1 & 2 & 1\\- & - & - & - & -\\- & - & - & - & -\end{Bmatrix}$$

Comentario:
Se podría atender a P_6

$$A=\begin{Bmatrix}5 & 2 & 2 & 3 & 3\end{Bmatrix}$$

$$C=\begin{Bmatrix}0 & 0 & 1 & 2 & 0\\0 & 1 & 0 & 0 & 1\\0 & 1 & 0 & 0 & 0\\0 & 0 & 0 & 1 & 1\\- & - & - & - & -\\- & - & - & - & -\\- & - & - & - & -\end{Bmatrix} \quad R=\begin{Bmatrix}3 & 3 & 2 & 2 & 3\\5 & 2 & 3 & 2 & 2\\5 & 3 & 0 & 5 & 2\\5 & 3 & 0 & 5 & 2\\- & - & - & - & -\\- & - & - & - & -\\- & - & - & - & -\end{Bmatrix}$$

Comentario:
Se podría atender a P_5 pero ya no quedarían suficientes recursos para atender a ningún proceso más. Al no poder encontrar un camino para que todos los procesos terminen entonces el estado es inseguro.
