

Examen 2015: El profesor Bacterio



En los laboratorios de química de la TIA se están realizando unas pruebas para producir agua artificialmente. El descubrimiento podría revolucionar la vida en la tierra tal y como la conocemos actualmente.

Como el presupuesto de la TIA es realmente limitado, su director ha contactado con nosotros para que simulemos cómo funcionaría la creación de agua a partir de sus componentes primigenios, Hidrógeno y Oxígeno.

La molécula de agua (H_2O) está formada por dos átomos de Hidrógeno(H) y un átomo de Oxígeno (O).

En la simulación, el sistema está generando continuamente agua a partir de la máquina productora, a la que llamaremos “Controlador”.

Esta máquina está conectada a dos tuberías “*tuberiaHidrogeno*” y “*tuberiaOxigeno*” que producen **infinitos** átomos () de cada uno de los dos elementos respectivamente.

La tubería de Hidrógeno pasa al “Controlador” átomos de Hidrógeno.

La tubería de Oxígeno pasa al “Controlador” átomos de Oxígeno.

Cada vez que se produce H_2O , el “Controlador” hace una espera de 300ms para evitar el recalentamiento.

Si el “Controlador” ya tiene suficientes átomos de un elemento para formar una molécula, no tomaría más átomos de ese mismo elemento hasta que cree una molécula de agua.



Realiza un programa que simule el comportamiento de este test científico. Para ello utiliza las directivas de sincronización “wait()”, “notify()” ó “notifyAll()” como consideres oportuno.

Pinta por pantalla:

- Cada vez que se tome un átomo por el “Controlador”. Indicando qué átomo se ha tomado.

- Cada vez que una tubería tenga que esperar porque ya hay átomos suficientes en el controlador. Indicando qué átomo es el que espera.

- Cada vez que se produce una molécula de agua

```
Se ha metido un oxígeno
Se ha metido un hidrógeno
No se pueden meter más oxígenos.
Se ha metido un hidrógeno
Se hizo agua
```

```
Se ha metido un oxígeno
Se ha metido un hidrógeno
No se pueden meter más oxígenos.
Se ha metido un hidrógeno
Se hizo agua
```

```
Se ha metido un hidrógeno
Se ha metido un hidrógeno
No se pueden meter más hidrógenos.
Se ha metido un oxígeno
Se hizo agua
```

```
Se ha metido un oxígeno
No se pueden meter más oxígenos.
Se ha metido un hidrógeno
Se ha metido un hidrógeno
Se hizo agua
```

El barbero dormilón

Una barbería está compuesta de una sala de espera, con n sillas, y la habitación del barbero con la silla correspondiente. Si no hay clientes, el barbero se duerme. Si un cliente entra en la sala de espera y todas las sillas están ocupadas, entonces abandona la barbería. Si el barbero está ocupado y hay sillas disponibles, entonces el cliente se sienta a esperar en una de las sillas. Si el barbero está dormido, el cliente le despierta.



Escribir un programa concurrente que coordine a los procesos barbero y clientes.

Recomendaciones:

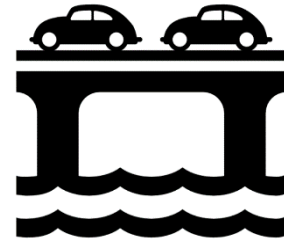
- Haz la simulación con 10 clientes y 5 sillas en la sala de espera.
- Si la sala de espera está llena el cliente lo vuelve a intentar pasados 4 segundos.
- Para hacer una espera más real, implementa el corte del pelo del cliente fuera del monitor. El barbero siempre tarda en cortar el pelo 5 segundos.

Posible ejecución del programa:

```
Barbero esperando cliente,.. Zzzzzzz
----El cliente 0 se sienta en la silla del Barbero.Sillas ocupadas: 0
----El cliente 0 despierta al Barbero
Barbero cortando el pelo XXXX
----El cliente 1 se sienta en la sala de espera. Sillas ocupadas: 1
Barbero termina de cortar el pelo
----El cliente 1 se sienta en la silla del Barbero.Sillas ocupadas: 0
----El cliente 2 se sienta en la sala de espera. Sillas ocupadas: 1
----El cliente 6 se sienta en la sala de espera. Sillas ocupadas: 2
----El cliente 3 se sienta en la sala de espera. Sillas ocupadas: 3
----El cliente 5 se sienta en la sala de espera. Sillas ocupadas: 4
----El cliente 4 se sienta en la sala de espera. Sillas ocupadas: 5
Sala de espera llena, el cliente 7 abandona la peluquería
Sala de espera llena, el cliente 9 abandona la peluquería
Sala de espera llena, el cliente 8 abandona la peluquería
Barbero cortando el pelo XXXX
Barbero termina de cortar el pelo
----El cliente 4 se sienta en la silla del Barbero.Sillas ocupadas: 4
Barbero cortando el pelo XXXX
Barbero termina de cortar el pelo
    ---El cliente 1 se va con el pelo cortado
    ---El cliente 0 se va con el pelo cortado
----El cliente 2 se sienta en la silla del Barbero.Sillas ocupadas: 3
Barbero cortando el pelo XXXX
Barbero termina de cortar el pelo
    ---El cliente 4 se va con el pelo cortado
----El cliente 5 se sienta en la silla del Barbero.Sillas ocupadas: 2
El cliente 8 se sienta en la sala de espera. Sillas ocupadas: 2
```

Sistema de seguridad en el puente

Se quiere desarrollar un sistema de seguridad que impida que un puente cargue más de 15.000 Kg y no haya simultáneamente más de 10 vehículos atravesándolo.



Cuando un vehículo quiere entrar en el puente, ejecuta el método *entrarPuente*. Cuando abandona el puente, ejecuta el método *salirPuente*. Ambos métodos tienen como argumento el peso del vehículo correspondiente. El primer método, además, tiene un argumento booleano que indica si el vehículo es una ambulancia. El perfil de estas operaciones:

... void *entrarPuente*(int numVehiculo, int peso, boolean esAmbulancia)

... void *salirPuente*(int numVehiculo, int peso, boolean esAmbulancia)

Un vehículo no recibirá permiso para entrar en el puente si dadas sus características o el estado del mismo, se incumplen los requisitos de seguridad. Además, las ambulancias tienen prioridad para acceder al puente respecto al resto de los vehículos. Es decir, si hay ambulancias esperando a entrar y llega un vehículo, cruzaría la ambulancia siempre y cuando se cumpla la seguridad del puente (peso y número de vehículos)

Se debe desarrollar un monitor que proporcione las dos operaciones previas según se han descrito y que garantice exclusión mutua en el acceso a los datos compartidos.

Recomendaciones:

- Los coches pesan entre 1.000 y 6.000 kilos y las ambulancias entre 4.000 y 8.000 kilos.
- Créate 10 coches y 5 ambulancias y arranca antes los coches que las ambulancias

Prueba el programa cambiando el número de vehículos y su peso.

```
Vehiculo 0 intentando entrar en el puente, peso: 5528 peso actual: 0 número de vehículos: 0
===Vehiculo 0 cruzando el puente
Vehiculo 5 intentando entrar en el puente, peso: 6194 peso actual: 5528 número de vehículos: 1
===Vehiculo 5 cruzando el puente
Vehiculo 6 intentando entrar en el puente, peso: 6262 peso actual: 11722 número de vehículos: 2
Vehiculo 4 intentando entrar en el puente, peso: 2133 peso actual: 11722 número de vehículos: 2
===Vehiculo 4 cruzando el puente
Vehiculo 3 intentando entrar en el puente, peso: 4908 peso actual: 13855 número de vehículos: 3
Vehiculo 8 intentando entrar en el puente, peso: 2515 peso actual: 13855 número de vehículos: 3
Vehiculo 2 intentando entrar en el puente, peso: 5082 peso actual: 13855 número de vehículos: 3
Vehiculo 7 intentando entrar en el puente, peso: 5931 peso actual: 13855 número de vehículos: 3
Vehiculo 9 intentando entrar en el puente, peso: 3285 peso actual: 13855 número de vehículos: 3
Ambulancia 0 intentando entrar en el puente, peso: 4469 peso actual: 13855 número de vehículos: 3
Ambulancia 2 intentando entrar en el puente, peso: 5620 peso actual: 13855 número de vehículos: 3
Ambulancia 3 intentando entrar en el puente, peso: 9258 peso actual: 13855 número de vehículos: 3
Vehiculo 1 intentando entrar en el puente, peso: 4096 peso actual: 13855 número de vehículos: 3
Ambulancia 4 intentando entrar en el puente, peso: 8153 peso actual: 13855 número de vehículos: 3
Ambulancia 1 intentando entrar en el puente, peso: 4848 peso actual: 13855 número de vehículos: 3
Vehiculo 0 saliendo del puente, peso actual del puente: 8327
===Ambulancia 1 cruzando el puente
Vehiculo 5 saliendo del puente, peso actual del puente: 6981
===Ambulancia 0 cruzando el puente
Vehiculo 4 saliendo del puente, peso actual del puente: 9317
===Ambulancia 2 cruzando el puente
Ambulancia 1 saliendo del puente, peso actual del puente: 10089
Ambulancia 0 saliendo del puente, peso actual del puente: 5620
===Ambulancia 4 cruzando el puente
Ambulancia 2 saliendo del puente, peso actual del puente: 8153
===Vehiculo 6 cruzando el puente
Vehiculo 6 saliendo del puente, peso actual del puente: 8153
===Vehiculo 1 cruzando el puente
===Vehiculo 8 cruzando el puente
Ambulancia 4 saliendo del puente, peso actual del puente: 6611
===Vehiculo 3 cruzando el puente
===Vehiculo 9 cruzando el puente
Vehiculo 1 saliendo del puente, peso actual del puente: 10708
Vehiculo 8 saliendo del puente, peso actual del puente: 8193
===Vehiculo 2 cruzando el puente
Vehiculo 3 saliendo del puente, peso actual del puente: 8367
Vehiculo 9 saliendo del puente, peso actual del puente: 5082
===Ambulancia 3 cruzando el puente
Vehiculo 2 saliendo del puente, peso actual del puente: 9258
Ambulancia 3 saliendo del puente, peso actual del puente: 0
===Vehiculo 7 cruzando el puente
Vehiculo 7 saliendo del puente, peso actual del puente: 0
```

La aldea Gala

En la aldea gala viven N galos. Estos galos beben de una gran marmita común que tiene una capacidad de M raciones del brebaje mágico. Cuando un galo quiere beber, se sirve él mismo de la marmita, a menos que esté vacía. Si la marmita está vacía, el galo despierta al druida *Panoramix* y espera a que el druida haya rellenado la marmita. Cuando lo ha hecho, y antes de dejar beber a otro galo, bebe él mismo.



Después de alguna otra actividad distinta, los galos vuelven a querer beber. El druida, por su parte, vuelve a dormir cuando ha rellenado la marmita.

Se pide desarrollar un monitor (MARMITA) con dos procedimientos, uno para que beba un galo (BEBER) y otro para rellenar la marmita (RELLENAR). La solución debe evitar interbloqueos y debe despertar al druida sólo cuando la marmita esté vacía.

Ejemplo de ejecución con 6 galos y una marmita con capacidad para 4 pociones:

```

---Asterix está tomando su ración de brebaje mágico. Raciones restantes: 3
---Falbalá está tomando su ración de brebaje mágico. Raciones restantes: 2
---Idéafix está tomando su ración de brebaje mágico. Raciones restantes: 1
---Asurancetúrix está tomando su ración de brebaje mágico. Raciones restantes: 0
000-La marmita está vacía, Obelix despierta a Panoramix
+++Panoramix ya ha rellenado la marmita, toma una ración. Raciones restantes: 3
---Abraracúrcix está tomando su ración de brebaje mágico. Raciones restantes: 2
---Obelix está tomando su ración de brebaje mágico. Raciones restantes: 1
---Asurancetúrix está tomando su ración de brebaje mágico. Raciones restantes: 0
000-La marmita está vacía, Idéafix despierta a Panoramix
+++Panoramix ya ha rellenado la marmita, toma una ración. Raciones restantes: 3
---Falbalá está tomando su ración de brebaje mágico. Raciones restantes: 2
---Asterix está tomando su ración de brebaje mágico. Raciones restantes: 1
---Idéafix está tomando su ración de brebaje mágico. Raciones restantes: 0
000-La marmita está vacía, Abraracúrcix despierta a Panoramix
+++Panoramix ya ha rellenado la marmita, toma una ración. Raciones restantes: 3
---Obelix está tomando su ración de brebaje mágico. Raciones restantes: 2
---Abraracúrcix está tomando su ración de brebaje mágico. Raciones restantes: 1
---Asterix está tomando su ración de brebaje mágico. Raciones restantes: 0
000-La marmita está vacía, Falbalá despierta a Panoramix
+++Panoramix ya ha rellenado la marmita, toma una ración. Raciones restantes: 3
---Abraracúrcix está tomando su ración de brebaje mágico. Raciones restantes: 2
---Obelix está tomando su ración de brebaje mágico. Raciones restantes: 1
---Asurancetúrix está tomando su ración de brebaje mágico. Raciones restantes: 0
000-La marmita está vacía, Asterix despierta a Panoramix
+++Panoramix ya ha rellenado la marmita, toma una ración. Raciones restantes: 3
---Idéafix está tomando su ración de brebaje mágico. Raciones restantes: 2
---Asterix está tomando su ración de brebaje mágico. Raciones restantes: 1
---Falbalá está tomando su ración de brebaje mágico. Raciones restantes: 0
000-La marmita está vacía, Abraracúrcix despierta a Panoramix
+++Panoramix ya ha rellenado la marmita, toma una ración. Raciones restantes: 3
---Asurancetúrix está tomando su ración de brebaje mágico. Raciones restantes: 2
---Obelix está tomando su ración de brebaje mágico. Raciones restantes: 1
---Abraracúrcix está tomando su ración de brebaje mágico. Raciones restantes: 0

```

El Club de Golf

El Club de Golf “El hoyo 8” únicamente permite jugar con los palos y pelotas que él mismo alquila. La cantidad de material disponible en el club es reducida: concretamente, 15 pelotas y 15 palos de golf. Como paso previo a jugar, cada socio del club debe solicitar el alquiler del material necesario. Hay dos clases de socios: novatos y



con experiencia. Los jugadores con experiencia siempre alquilan una única pelota y entre

2 y 5 palos para jugar. Un jugador novato suele alquilar un número mayor de pelotas (1 a 5 pelotas) y dos únicos palos.

Se pide desarrollar una aplicación que simule la actividad que un grupo de jugadores lleva a cabo en las instalaciones del club.

Concretamente, habrá 6 jugadores con experiencia y otros 6 novatos. Cada jugador se identificará a través de un número entero comprendido entre 1 y 12, seguido de un signo ‘+’ si se trata de un jugador con experiencia y de un signo ‘-’ si se trata de un jugador novato.

Cada jugador muestra, repetidamente, el siguiente comportamiento, independientemente de su nivel de experiencia: decide jugar y reserva el material necesario, juega durante un tiempo (sleep (10000)), devuelve el material que ha usado. En ningún caso podrá comenzar a jugar si no dispone del material necesario, el número de jugadores que pueden jugar a la vez en el campo es ilimitado. Cada jugador jugará únicamente un partido.

Para entender cuál es el funcionamiento del sistema implementado, cada jugador deberá escribir en la pantalla mensajes indicando su comportamiento:

- Antes de realizar una reserva deberá escribir un mensaje donde muestre su identidad y el número de recursos solicitados.
- Una vez haya conseguido reservar el material que necesita, escribirá un mensaje indicando su identidad, los recursos alquilados y su intención de comenzar a jugar.
- Cuando vaya a realizar la devolución deberá escribir un mensaje donde muestre su identidad y los recursos devueltos.

En cuanto al formato de los mensajes, la representación de la identidad de un jugador y el material que reserva o va a devolver se ajustará al siguiente formato: “*identidad, acción que realiza [número de palos, números de pelotas]*”.

Posible salida por pantalla:


```

Jugador:1- quiere jugar al golf, necesita [2,2]. Hay 15 palos y 15 pelotas
Jugador:1- alquila material[2,2]
    Jugador:1- está jugando al golf
Jugador:10+ quiere jugar al golf, necesita [5,1]. Hay 13 palos y 13 pelotas
Jugador:10+ alquila material[5,1]
    Jugador:10+ está jugando al golf
Jugador:9- quiere jugar al golf, necesita [2,3]. Hay 8 palos y 12 pelotas
Jugador:9- alquila material[2,3]
    Jugador:9- está jugando al golf
Jugador:8+ quiere jugar al golf, necesita [2,1]. Hay 6 palos y 9 pelotas
Jugador:8+ alquila material[2,1]
    Jugador:8+ está jugando al golf
Jugador:7- quiere jugar al golf, necesita [2,1]. Hay 4 palos y 8 pelotas
Jugador:7- alquila material[2,1]
    Jugador:7- está jugando al golf
Jugador:6+ quiere jugar al golf, necesita [2,1]. Hay 2 palos y 7 pelotas
Jugador:6+ alquila material[2,1]
    Jugador:6+ está jugando al golf
Jugador:5- quiere jugar al golf, necesita [2,1]. Hay 0 palos y 6 pelotas
xxxJugador:5-,no puede alquilar material.Actualmente hay 0 palos y 6 pelotas
Jugador:4+ quiere jugar al golf, necesita [2,1]. Hay 0 palos y 6 pelotas
xxxJugador:4+,no puede alquilar material.Actualmente hay 0 palos y 6 pelotas
Jugador:3- quiere jugar al golf, necesita [2,3]. Hay 0 palos y 6 pelotas
xxxJugador:3-,no puede alquilar material.Actualmente hay 0 palos y 6 pelotas
Jugador:2+ quiere jugar al golf, necesita [6,1]. Hay 0 palos y 6 pelotas
xxxJugador:2+,no puede alquilar material.Actualmente hay 0 palos y 6 pelotas
Jugador:0+ quiere jugar al golf, necesita [3,1]. Hay 0 palos y 6 pelotas
xxxJugador:0+,no puede alquilar material.Actualmente hay 0 palos y 6 pelotas
Jugador:11- quiere jugar al golf, necesita [2,4]. Hay 0 palos y 6 pelotas
xxxJugador:11-,no puede alquilar material.Actualmente hay 0 palos y 6 pelotas
    Jugador:1-terminó el partido, devuelve el material[2,2]
Jugador:11- alquila material[2,4]
    Jugador:11- está jugando al golf
        Jugador:10+terminó el partido, devuelve el material[5,1]
Jugador:5- alquila material[2,1]
    Jugador:5- está jugando al golf
Jugador:4+ alquila material[2,1]
    Jugador:4+ está jugando al golf
        Jugador:7-terminó el partido, devuelve el material[2,1]
        Jugador:8+terminó el partido, devuelve el material[2,1]
Jugador:0+ alquila material[3,1]
    Jugador:0+ está jugando al golf
Jugador:3- alquila material[2,3]
    Jugador:3- está jugando al golf
        Jugador:9-terminó el partido, devuelve el material[2,3]
        Jugador:6+terminó el partido, devuelve el material[2,1]
        Jugador:11-terminó el partido, devuelve el material[2,4]
Jugador:2+ alquila material[6,1]
    Jugador:2+ está jugando al golf
        Jugador:5-terminó el partido, devuelve el material[2,1]
        Jugador:4+terminó el partido, devuelve el material[2,1]
        Jugador:3-terminó el partido, devuelve el material[2,3]
        Jugador:0+terminó el partido, devuelve el material[3,1]
        Jugador:2+terminó el partido, devuelve el material[6,1]

```

La tienda de pájaros

En una tienda de pájaros tenemos una jaula con 10 canarios que revolotean en su interior. Regularmente (cada 10ms) cada canario quiere comer de un comedero de alpiste, en el que solamente puede haber 3 canarios comiendo al mismo tiempo. Si un canario quiere comer y el comedero está ocupado, se debe esperar a que haya hueco. Por su parte, el encargado de la tienda de vez en cuando (cada 10ms) repone el alpiste del comedero. Mientras el encargado está reponiendo, ningún canario puede estar comiendo: el encargado debe esperar a que no haya canarios en el comedero y, una vez libre, ningún canario entra a comer hasta que el encargado termina de reponer.



Realiza un algoritmo para que los canarios y el encargado se sincronicen entre todos ellos conforme al enunciado expuesto.

Los canarios comen (tardan 2sg) fuera del bloque sincronizado para liberar el candado del monitor y que puedan entrar más canarios a la sección crítica.

Posible salida por pantalla:

```
-----Canario 1 está comiendo alpiste <0>
-----Canario 2 está comiendo alpiste <0>
-----Canario 0 está comiendo alpiste <0>
El encargado quiere reponer alpiste
    Canario 2 abandona el comedero WW_O_WW
    Canario 1 abandona el comedero WW_O_WW
    Canario 0 abandona el comedero WW_O_WW
+++++Reponiendo alpiste
-----Canario 8 está comiendo alpiste <0>
-----Canario 7 está comiendo alpiste <0>
-----Canario 9 está comiendo alpiste <0>
    Canario 8 abandona el comedero WW_O_WW
-----Canario 3 está comiendo alpiste <0>
    Canario 9 abandona el comedero WW_O_WW
    Canario 7 abandona el comedero WW_O_WW
-----Canario 5 está comiendo alpiste <0>
-----Canario 6 está comiendo alpiste <0>
El encargado quiere reponer alpiste
    Canario 3 abandona el comedero WW_O_WW
    Canario 5 abandona el comedero WW_O_WW
    Canario 6 abandona el comedero WW_O_WW
+++++Reponiendo alpiste
-----Canario 0 está comiendo alpiste <0>
-----Canario 1 está comiendo alpiste <0>
-----Canario 2 está comiendo alpiste <0>
    Canario 2 abandona el comedero WW_O_WW
```

Servidor de Descargas

Disponemos en el Instituto de un servidor de descargas de archivos. Cada archivo almacenado en el sistema tiene un nombre (String) y un tamaño (float) comprendido entre 300MB y 5 GB (5120MB). Para descargar cada Usuario llamará al método **descargar** que recibirá como parámetro el nombre del usuario y el archivo a descargar, el tiempo de descarga de nuestro servidor es siempre de 1MB□1ms. El sistema tiene una restricción, y es que no puede haber descargas simultáneas que sumen más de 10 GB. Si un usuario solicita la descarga de un archivo y su posible descarga sobrepasa el límite, quedará esperando hasta que haya suficiente ancho de banda.



Diseño un programa para simular el servidor de descargas que cumpla con el límite de descarga. Para probarlo utiliza los siguientes datos:

- En nuestro servidor se alojan 10 archivos, nombrados de la forma: archivo-X
- Tendremos 4 usuarios (David, María, Sergio, Marta) que intentarán la descarga aleatoria de 2 archivos del servidor.

Posible salida por pantalla:

```
El usuario David desea descargarse el archivo: archivo-0 de 4895.548MB
==>Se está descargando el archivo.....archivo-0(David) de 4895.548MB
El usuario Maria desea descargarse el archivo: archivo-7 de 2171.3813MB
==>Se está descargando el archivo.....archivo-7(Maria) de 2171.3813MB
El usuario Marta desea descargarse el archivo: archivo-4 de 3504.7334MB
El usuario Sergio desea descargarse el archivo: archivo-5 de 2103.0913MB
XXActualmente se están descargando simultaneamente 7066.929MB. Usuario: Marta esperando...
==>Se está descargando el archivo.....archivo-5(Sergio) de 2103.0913MB
    Archivo archivo-5(Sergio) descargado.....
El usuario Sergio desea descargarse el archivo: archivo-7 de 2171.3813MB
XXActualmente se están descargando simultaneamente 7066.929MB. Usuario: Marta esperando...
==>Se está descargando el archivo.....archivo-7(Sergio) de 2171.3813MB
    Archivo archivo-7(Maria) descargado.....
XXActualmente se están descargando simultaneamente 7066.929MB. Usuario: Marta esperando...
El usuario Maria desea descargarse el archivo: archivo-8 de 925.68427MB
==>Se está descargando el archivo.....archivo-8(Maria) de 925.68427MB
    Archivo archivo-8(Maria) descargado.....
XXActualmente se están descargando simultaneamente 7066.929MB. Usuario: Marta esperando...
    Archivo archivo-7(Sergio) descargado.....
==>Se está descargando el archivo.....archivo-4(Marta) de 3504.7334MB
    Archivo archivo-0(David) descargado.....
El usuario David desea descargarse el archivo: archivo-4 de 3504.7334MB
==>Se está descargando el archivo.....archivo-4(David) de 3504.7334MB
    Archivo archivo-4(Marta) descargado.....
El usuario Marta desea descargarse el archivo: archivo-1 de 2690.8135MB
==>Se está descargando el archivo.....archivo-1(Marta) de 2690.8135MB
    Archivo archivo-4(David) descargado.....
    Archivo archivo-1(Marta) descargado.....
```


Estación de servicio

En una estación de servicio existen tres surtidores, uno para gasolina, otro para gasoil y otro para biocombustible. En cada surtidor se puede atender a un solo vehículo en cada instante, estando ubicados en vías de capacidad finita para los vehículos en espera, específicamente la vía para la gasolina tiene capacidad CG vehículos, la del gasoil CD y la del biocombustible CB. El repostaje es realizado por el propietario del vehículo, de forma que cuando un vehículo llega a la estación este solicita el repostar (*solicitar_repostaje*) mediante una petición en la que especifica el tipo de combustible que necesita, si la vía asociada al tipo de combustible requerido está llena, entonces debe irse inmediatamente, en caso contrario deberá esperar si otro vehículo está repostando o repostar inmediatamente si no hay vehículo repostando. Al terminar informa de la finalización del repostaje (*finalizar_repostaje*). Los vehículos deben ser procesos concurrentes.



Utilizando monitores desarrolla una solución que permita el repostaje de los vehículos cumpliendo los requisitos expresados.

Ejemplo de ejecución utilizando los siguientes valores:

- 12 coches cuyo tipo de combustible se elige al azar.
- CG = 3; CD = 2; CB = 1

Posible salida por pantalla:

```
Coche-1 desea repostar DIESEL
  Coche-1 está repostando DIESEL
Coche-2 desea repostar BIOCOMBUSTIBLE
  Coche-2 está repostando BIOCOMBUSTIBLE
Coche-0 desea repostar BIOCOMBUSTIBLE
El surtidor de BIOCOMBUSTIBLE está ocupado,Coche-0 espera en la cola [1,1]
Coche-3 desea repostar GASOLINA
  Coche-3 está repostando GASOLINA
Coche-4 desea repostar DIESEL
El surtidor de DIESEL está ocupado,Coche-4 espera en la cola [1,2]
Coche-5 desea repostar GASOLINA
Coche-6 desea repostar BIOCOMBUSTIBLE
  XXX-La cola de BIOCOMBUSTIBLE está llena, Coche-6 abandona la estación
El surtidor de GASOLINA está ocupado,Coche-5 espera en la cola [1,3]
Coche-7 desea repostar DIESEL
El surtidor de DIESEL está ocupado,Coche-7 espera en la cola [2,2]
Coche-10 desea repostar DIESEL
Coche-11 desea repostar BIOCOMBUSTIBLE
Coche-9 desea repostar BIOCOMBUSTIBLE
Coche-8 desea repostar DIESEL
  XXX-La cola de BIOCOMBUSTIBLE está llena, Coche-9 abandona la estación
  XXX-La cola de DIESEL está llena, Coche-8 abandona la estación
  XXX-La cola de DIESEL está llena, Coche-10 abandona la estación
  XXX-La cola de BIOCOMBUSTIBLE está llena, Coche-11 abandona la estación
=>Coche-1 terminó de repostar DIESEL abandona la estación
Coche-7 está repostando DIESEL
=>Coche-2 terminó de repostar BIOCOMBUSTIBLE abandona la estación
Coche-0 está repostando BIOCOMBUSTIBLE
=>Coche-3 terminó de repostar GASOLINA abandona la estación
Coche-5 está repostando GASOLINA
=>Coche-7 terminó de repostar DIESEL abandona la estación
Coche-4 está repostando DIESEL
=>Coche-0 terminó de repostar BIOCOMBUSTIBLE abandona la estación
=>Coche-5 terminó de repostar GASOLINA abandona la estación
=>Coche-4 terminó de repostar DIESEL abandona la estación
```