1.– **El puente de Puente Duero**. El pueblo de Puente Duero debe su nombre a un viejo puente romano sobre el conocido río. El puente es estrecho y no permite el cruce de dos coches que circulen en sentidos contrarios. Por su peculiar forma, alto en el centro y bajo en los extremos, no hay visibilidad, y desde un lado no podemos saber si hay un coche empezando a cruzar el puente por el otro.

Se han instalado cuatro sensores y dos barreras. Dos sensores indican la llegada de coches en cada dirección, emitiendo una señal por cada coche que pasa, y los otros dos indican salidas respectivas. Las barreras bloquean las entradas al puente, y cuando reciben una señal se levantan para permitir el paso de un coche (y uno sólo); acto seguido se vuelven a cerrar. Diseñe un programa concurrente, incluyendo un proceso que emplee las señales recibidas de los sensores para emitir señales apropiadas a las barreras, de manera que:

1) Nunca crucen simultáneamente el puente vehículos en sentidos opuestos.
2) Nadie espere eternamente para pasar
3) Todos pasen lo más pronto posible salvo violación de las normas 1) o 2).

Las señales de los sensores han de atenderse inmediatamente (o al menos lo más pronto posible), para no "perder" el paso de ningún coche.

2.– **El Problema del Río Africano**. En el África profunda viven junto a un río una tribu de caníbales y un grupo de misioneros. El río se puede cruzar mediante un bote, con capacidad para tres personas.

Hechos a tener en cuenta:

a) Como las aguas son bravas, el bote siempre tiene que transportar carga completa.
b) Si se sientan dos misioneros y un caníbal, éstos le comen el tarro al caníbal, hasta el punto de que el caníbal se arroja al agua: hay que evitar esta situación. Las demás combinaciones no dan problemas.
c) Una persona se encuentra siempre en una de las dos orillas del río, y de vez en cuando intenta cruzar el río.
d) La persona ha de esperarse hasta que el barco esté en la misma orilla que la persona, y que exista una combinación de pasajeros adecuada.

Construya un programa concurrente para simular el sistema.

3.– En el **patio de una casa** se encuentran cuatro niños que continuamente efectúan las siguientes tareas. Inicialmente se encuentran jugando. Cuando un niño tiene hambre recurre a un plato y toma una porción de queso del mismo. Acto seguido duerme durante 30 unidades de tiempo y a continuación vuelve a sus juegos. Este ciclo se repite indefinidamente.

Si cuando un niño intenta comer, encuentra el plato vacío, avisa a la madre de este suceso, esperando entonces a que la madre ponga comida en el plato.

La madre, entre otras cosas, se encarga de realizar las siguientes dos tareas:

i) Despertar a los niños que, después de comer, han dormido 30 unidades de tiempo (los niños son muy dormilones y no se despertarían de otra forma).

ii) Reponer el plato, cuando esté vacío, con cuatro porciones de queso, si es que algún niño solicita comida.

Implemente un programa en el que existan cuatro procesos "niño" y un proceso "madre", que actúen siguiendo el comportamiento anteriormente descrito. La solución propuesta debe estar libre de interbloqueo e inanición.

4.– Cintas transportadoras

La nueva terminal de Barajas estará conectada con la actual mediante un par de cintas transportadoras (una en cada sentido) para hacer menos fatigoso el desplazamiento de los viajeros. Las cintas poseen un mecanismo de control para que estén en marcha solo cuando se necesitan. Esto se consigue mediante unos sensores a la entrada de cada cinta que detectan la llegada de usuarios.

El criterio seguido es:

Si una cinta esta parada, debe ponerse en marcha al detectarse la presencia de un usuario.

Si una cinta está en marcha, se para si pasa un minuto sin que llegue nadie.

Para implementar este sistema se dispone de:

- un procedimiento de acceso al sensor
- procedimientos de manejo del motor que ponen en marcha y detienen la cinta N, respectivamente, y
- un único proceso Reloj que cada segundo envia un mensaje a través del canal.

Además, resulta que si surge algún problema en el motor de una de las cintas se requiere que al menos la otra cinta siga funcionando.

Extra: ¿Qué modicaciones habría que realizar en vuestro sistema si ahora queremos que en vez de tener prefijados los sentidos de la marcha, estos sean variables? Se supone que disponemos de sensores en ambos extremos de cada cinta y el requisito adicional sería que las cintas no deben nunca desplazarse en el mismo sentido.

5.– **Club de Submarinismo**. Un club de submarinismo dispone de un conjunto de *B* botellas de aire comprimido para alquilar. *N* grupos de buceadores, que se representarán como procesos, solicitan al club un número variable de botellas, que se garantiza es siempre menor o igual que *B*, para realizar una inmersión. Cada grupo devolverá las botellas al club al finalizar la inmersión. Un grupo puede realizar cualquier número de inmersiones consecutivas.

Diseña un programa concurrente compuesto por los *N* procesos y un gestor, que satisfaga las siguientes propiedades:

a) Cada botella ha de utilizarse en exclusión mutua, es decir, sólo un grupo a la vez puede disponer de ella.

b) El sistema estará libre de bloqueos e inaniciones.

c) Se ha de procurar usar al máximo las botellas del club, ya que éste está planteado como un negocio lucrativo.

6.– **El Camino de Santiago.** Distintos peregrinos, cada uno con unas fuerzas y dinero diferentes van a hacer el Camino de Santiago. Los peregrinos pueden partir desde tres diferentes salidas (Roncesvalles, Somport, o bien empezar en donde se unen estos dos caminos, Nájera) pero el camino es largo y deberán realizar distintas paradas para dormir y reponer fuerzas.

Cuando uno de los peregrinos llega a una ciudad lo primero que hace es ir a un albergue gratuito, para reponer sus fuerzas, pero éstos son de plazas limitadas, así que si están ocupadas todas las plazas irá a un hotel de la localidad. Este hotel le cobrará una cantidad determinada de dinero. Además, aquí descansará mejor y recuperará bastantes más puntos de fuerza que en el albergue. Si no dispone de pasta, tendrá que comprobar si todavía le queda suficiente fuerza para ir al siguiente pueblo, en cuyo caso irá de inmediato a la siguiente ciudad. Si a parte de no tener dinero, tampoco tiene fuerzas no le quedará más remedio que abandonar.

También será motivo de abandono el no tener fuerzas suficientes para ir al siguiente punto después de haberse recuperado en un albergue u hotel.

Cuando uno termina de recuperarse en una ciudad, y de ella salen varios caminos, si tiene suficientes puntos de fuerza, intentará ir por el camino que más lejos le lleve. Si un peregrino llega a Santiago se recuperará y pondrá fin a su viaje.

Cada ciudad esta separada de otra con distancias diferentes, por lo cual el consumo de fuerzas de cada peregrino irá en función de esta distancia.

Diseñe un programa que simule el sistema. Indique además, las modificaciones necesarias para resolver el problema teniendo presente la leyenda que hay en cada albergue: "*Deja lo que puedas, toma lo que necesites*".

OBSERVACIONES:

Problema 1: Todas Herramientas

Problemas 2 y 3: Java / Ada: Tipos Protegidos y Citas

Resto: Java (2) / Ada: Tipos Protegidos (2) y Citas (2).