

Cuadrúpedo Robotizado

La Agencia Nacional de Seguridad Española ha contactado con nosotros para que le ayudemos en el desarrollo de su nuevo prototipo cuadrúpedo.

Las condiciones que debe cumplir ese cuadrúpedo son las siguientes:



- Cada pata se controla con un hilo.
- Las cuatro patas se mueven por turnos. Cada pata sólo se puede mover una vez en cada turno. Es decir, al principio, cualquier pata se puede mover, pero ésta tiene que esperar hasta que se muevan las otras tres.
- Al principio de cada turno cualquier pata es candidata de moverse.
- Entre turno y turno hay que realizar una espera de un segundo.

Pinta por pantalla el movimiento de cada pata y marca el final de cada turno.

Ejemplo de salida por pantalla:

```
Se mueve la pata 0
Se mueve la pata 2
Se mueve la pata 3
Se mueve la pata 1
FIN DE TURNO
Se mueve la pata 3
Se mueve la pata 2
Se mueve la pata 0
Se mueve la pata 1
FIN DE TURNO
Se mueve la pata 1
Se mueve la pata 2
Se mueve la pata 3
Se mueve la pata 0
FIN DE TURNO
Se mueve la pata 2
Se mueve la pata 3
Se mueve la pata 1
Se mueve la pata 0
FIN DE TURNO
Se mueve la pata 0
Se mueve la pata 3
Se mueve la pata 2
Se mueve la pata 1
FIN DE TURNO
Se mueve la pata 2
```

Los filósofos



Cinco filósofos alrededor de una mesa pasan su vida comiendo o pensando. Cada filósofo tiene un plato de arroz y un palillo a la izquierda de su plato. Cuando un filósofo quiere comer, cogerá los dos palillos de cada lado del plato y comerá.

El problema es desarrollar el algoritmo que permita comer a los filósofos. Este algoritmo debe satisfacer la exclusión mutua (dos filósofos no pueden emplear el mismo palillo a la vez), además de evitar el interbloqueo y la inanición.

NOTAS:

Cada filósofo compite con los de su izquierda y su derecha para coger el palillo que separa a ambos. Es imposible que el mismo palillo sea utilizado por dos filósofos a la vez.

Todos los filósofos deben comer de vez en cuando, para que no mueran de hambre.

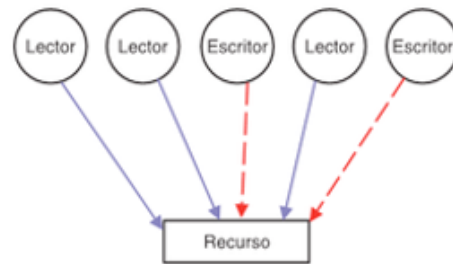
Cuidado con la situación en la que todos los filósofos tomen su palillo de la izquierda.
¡BLOQUEO!

```
Socrates coge los palillos.  
Socrates está comiendo  
Socrates suelta los palillos.  
    kant coge los palillos.  
    kant está comiendo  
    kant suelta los palillos.  
        Locke coge los palillos.  
        Locke está comiendo  
        Locke suelta los palillos.  
            Aristoteles coge los palillos.  
            Aristoteles está comiendo  
            Aristoteles suelta los palillos.  
Socrates coge los palillos.  
Socrates está comiendo  
    Locke coge los palillos.  
Socrates suelta los palillos.  
        Hegel coge los palillos.  
        Hegel está comiendo  
        Hegel suelta los palillos.  
    Locke está comiendo  
    Locke suelta los palillos.  
    kant coge los palillos.  
    kant está comiendo  
        Aristoteles coge los palillos.  
        Aristoteles está comiendo  
        Aristoteles suelta los palillos.  
    kant suelta los palillos.  
Socrates coge los palillos.  
Socrates está comiendo  
    Aristoteles coge los palillos.  
Socrates suelta los palillos.  
    Aristoteles está comiendo  
        Aristoteles suelta los palillos.  
            Hegel coge los palillos.  
            Hegel está comiendo  
            Hegel suelta los palillos.  
    Locke coge los palillos.  
    Locke está comiendo  
    Locke suelta los palillos.  
    kant coge los palillos.  
    kant está comiendo  
    kant suelta los palillos.  
        Hegel coge los palillos.  
        Hegel está comiendo  
        Hegel suelta los palillos.
```

Lectores y escritores

Existe un área de datos compartida entre una serie de procesos, algunos solamente leen los datos (lectores) y algunos solamente escriben los datos (escritores). Se nos pide satisfacer las siguientes condiciones:

- Cualquier número de lectores puede leer el recurso simultáneamente.
- El recurso sólo puede ser escrito por un escritor simultáneamente.
- Si un escritor está accediendo al recurso. Ningún lector puede leerlo.



Controla el área de exclusión mutua para que se satisfagan estas condiciones.

Vamos a realizar las simulaciones como si hubiera 3 lectores y 2 escritores.

El recurso sobre el que trabajan va a ser un String. Al inicio de la ejecución este recurso tiene el valor "vacío".

Cada escritor escribe 5 veces en su ejecución. Escribirá en el recurso su nombre + " --" +Vuelta por la que va en el bucle de escritura".

Por ejemplo, el escritor 2 en su cuarta ejecución del bucle escribirá en el recurso:

"Escritor2--4"

Cada lector lee 5 veces el recurso. Cada vez que lee el recurso pinta por pantalla:

LectorX --> "+"Valor del recurso"

Por ejemplo, si lee el lector 1 el recurso antes de que haya escrito ningún escritor:

"Lector1 --> vacío"

Si leyera después, teniendo en cuenta que el recurso valiera:

"Escritor2--4"

Pintaría por pantalla: "Lector1 --> Escritor2--4"

```
Escritores1--1
Escritores1--2
Escritores1--3
    Lectores1--1 --> Escritores1--3
    Lectores2--1 --> Escritores1--3
    Lectores3--1 --> Escritores1--3
Escritores2--1
Escritores2--2
Escritores2--3
Escritores2--4
Escritores2--5
Escritores1--4
Escritores1--5
    Lectores1--2 --> Escritores1--5
    Lectores2--2 --> Escritores1--5
    Lectores3--2 --> Escritores1--5
    Lectores1--3 --> Escritores1--5
    Lectores2--3 --> Escritores1--5
    Lectores3--3 --> Escritores1--5
    Lectores2--4 --> Escritores1--5
    Lectores3--4 --> Escritores1--5
    Lectores1--4 --> Escritores1--5
    Lectores3--5 --> Escritores1--5
    Lectores1--5 --> Escritores1--5
    Lectores2--5 --> Escritores1--5
```