

1. Codifica un programa que lance una carrera de 8 corredores, representados cada uno de ellos por un hilo:
 - a. Cada corredor tendrá un número de dorsal desde el 1 hasta el 8 y tendrá que incrementar una variable de 1 en 1 desde el 1 hasta el 1000 (en un bucle *for* por ejemplo) para llegar a la meta.
 - b. Los corredores saldrán en el mismo momento en el que se ejecute la aplicación
 - c. Cuando hayan llegado los 8 corredores se mostrará un listado de resultados.

Un ejemplo de ejecución podría ser el siguiente:

Carrera en curso....
Resultados:
Posición 1 corredor 5.
Posición 2 corredor 3.
Posición 3 corredor 7.
Posición 4 corredor 1.
Posición 5 corredor 2.
Posición 6 corredor 6.
Posición 7 corredor 4.
Posición 8 corredor 8.

Ejemplo código de parte del Main:

```
public static final int NUM_CORREDORES = 8;
public static void main(String[] args)
{

    Lista lista = new Lista(NUM_CORREDORES);

    Thread[] thCorredores = new Thread[NUM_CORREDORES];

    // Creación de los hilos
    for (int i = 0; i < thCorredores.length; i++)
    {
        thCorredores[i] = new Thread(new RnCorredor(i + 1, lista));
    }
    // Ejecución de los hilos
    for (int i = 0; i < thCorredores.length; i++)
    {
        thCorredores[i].start();
    }
}
```

Ejemplo parte del código de la clase Lista:

```
public class Lista
{
    private int[] lista;
    private int posicion;
    public Lista(int numCorredores)
    {
        lista = new int[numCorredores];
        posicion = 0;
    }
    public synchronized void apuntarCorredor(int dorsal)
```

Ejemplo parte código clase RnCorredor

```
public class RnCorredor implements Runnable
{
    private int numCorredor;
    private Lista lista;
    public RnCorredor(int numCorredor, Lista lista)
    {
```

2. Sincroniza 2 hilos, uno encargado de escribir ping y otro encargado de escribir pong, para que el resultado de la ejecución sea la secuencia infinita ping pong ping pong...

```
public class Main
{
    public static void main(String[] args)
    {
        EscribirPingPong escribePingPong = new EscribirPingPong();
        Thread thPing = new Thread(new EscribirPing(escribirPingPong));
        Thread thPong = new Thread(new EscribirPong(escribirPingPong));
        thPong.start();
        thPing.start();
    }
}
```

```
public class EscribirPingPong
{
    public synchronized void escribirPing()
    {
        ...
        System.out.println("ping");
        ...
    }
    public synchronized void escribirPong()
    {
        ...
        System.out.println("pong");
        ...
    }
}
```