

1. Haga un programa en Java que genere 4 hilos usando la interfaz Runnable. El primer hilo debe contar del 1 al 50, el segundo del 51 al 100, el tercero del 101 al 150 y el cuarto del 151 al 200. La salida del programa debe ser similar a:

```
Hilo 3 101
Hilo 2 51
Hilo 4 151
Hilo 4 152
Hilo 4 153
Hilo 4 154
```

2. Repita el ejercicio anterior de forma que en lugar de usar la interfaz Runnable, se extienda de la clase Thread.
3. Haga una clase a la que se le pase un nombre de fichero y un String con los contenidos del archivo y la clase escriba el archivo en disco de forma asíncrona.
4. Haga una clase que se le pase un nombre de fichero y lea el fichero de forma asíncrona. Añada un método getContenidos() que devuelva el contenido del archivo leído. Dentro del método getContenidos() se debe bloquear el hilo que lo invoque hasta que los contenidos del archivo hayan sido leídos completamente.
5. Como aplicación de los dos ejercicios anteriores de la lectura y escritura asíncronas haga un programa que lea un archivo plantilla de la forma:

```
<!doctype html>

<html>
<head>
    <meta charset='utf-8'>
    <title>Suma del 1 al 100</title>
</head>
<body>
    <h1>La suma es {suma}</h1>
    El resultado de la suma de 1 a 100 es {suma}.
</body>
</html>
```

Mientras lo lea debe hacer la suma de los números del 1 al 100. Mientras escribe en pantalla el resultado de la suma, debe guardar en disco la plantilla reemplazando “{suma}” por el valor de la suma.

6. El método sleep de la clase Thread permite que un hilo se duerma durante el tiempo que se le pase como argumento. Lea la documentación del método Thread y haga un ejemplo de uso del método sleep.
7. Haga una clase, llamada Mensaje que contenga un número y dos métodos que permitan incrementar en uno o decrementar (restar uno). Haga dos hilos que accedan al mismo objeto. El hilo1 debe incrementar y el hilo2 decrementar. Después de incrementar o decrementar, cada hilo debe dormir, usando sleep, un tiempo aleatorio. Determine cuál es la sección crítica en este código y use la palabra clave de Java synchronized para que no se produzca una condición de carrera.
8. Repita el ejercicio anterior usando semáforos.
9. Se desea simular la siguiente situación usando hilos y semáforos: Un supermercado tiene 10 cajas funcionando. Hay 100 clientes que desean usar las cajas. Los clientes que desean usar las cajas entran a las mismas hasta que todos los puestos están ocupados, entonces forman una cola. Según se van liberando cajas, van entrando los clientes de la cola.