

Examen PSP

Información general:

<https://github.com/JorgeDuenasLerin/docencia-23-24/>

Todo lo que no cumpla los requisitos o no sea entregado según formato no será evaluado.

Entrega:

apellido1-apellido-nombre

```
\- trimestre1
    \- ejercicio1
    \- ejercicio2
\- trimestre2
    \- ejercicio1
    \- ejercicio2
```

Tiempo: 2:30 por trimestre.

Colisión en DNI

La letra de un DNI es un carácter de control de errores, puede ser calculado con el número de DNI, la operación módulo y el siguiente array.

```
String[] letras = {"T", "R", "W", "A", "G", "M", "Y", "F", "P", "D",
    "X", "B", "N", "J", "Z", "S", "Q", "V", "H", "L", "C", "K", "E"};
```

Por ejemplo 00000046 daría la 'T'.

Por ejemplo 00000047 daría la 'R'.

```
printf("%08d", dni); // Para pintar ceros a la izquierda.
```

Crea un programa en C que reciba por parámetro una letra de DNI y un número. El programa escribirá todos los DNIs con esa letra utilizando n procesos. Siendo n el segundo número pasado por parámetro. Además cuando un DNI sea un número primo escribirá un asterisco al lado del DNI

El programa principal esperará a sus hijos.

Pista: La mejor forma de distribuir el trabajo es con el módulo y el número de proceso dentro del bucle.

Puntos:

- 1.5 punto. Distribuir trabajo.
- 1.5 punto. Calcular y escribir DNIs en cada proceso.
- 1.0 punto. Escribe primo
- 1.0 punto. Padre espera a los hijos.

Threads

Retrete para 5 con control de acceso.

En un baño pueden entrar 5 personas como máximo. Una vez que entre una persona (Thread), escribirá “Entra thread ”, estará un tiempo aleatorio entre 10 seg y 60 seg esperando y escribirá “Terminé”. Cuando ya hay 5 personas en el baño si una más quiere entrar se quedará esperando. Cuando una persona sale avisa por si alguien más está esperando.

Crear las clases necesarias y un programa main que genere 20 threads que quieren entrar al baño.

Clases:

- Bano: con región crítica, conteo de gente y métodos entrar y salir
- Persona: con acceso a Bano, un número identificativo. En su `run` hace: entrar, sleep (simula el usar el baño) y salir
- Principal: Crea 20 personas y comienza sus threads

Puntos:

- 1 punto. Simular entrada y salida concurrente.
- 1 punto. Mecanismo de control de acceso (wait en la capacidad máxima)
- 1 punto. Ocupar el baño el tiempo indicado
- 1 punto. Notificar baño libre (notify)
- 1 punto. Entrar baño libre.

HTTP

Crea un servicio multithread para escribir n veces una palabra. Las url serán:

```
/hola/10/
/pacooooo/3/
```

Con estas peticiones se generará un HTML, la primera vez que se escriba la palabra estará en un `h1`, la segunda en un `h2`, así hasta el `h6`, las siguientes aparecerán en `p`. Por ejemplo `/pacooooo/10/`, mostrará

```
<h1>pacoooooooo</h1>
<h2>pacoooooooo</h2>
<h3>pacoooooooo</h3>
<h4>pacoooooooo</h4>
<h5>pacoooooooo</h5>
<h6>pacoooooooo</h6>
<p>pacoooooooo</p>
<p>pacoooooooo</p>
<p>pacoooooooo</p>
<p>pacoooooooo</p>
```

El sistema tendrá un log `peticiones.txt` (**ruta relativa** a la ejecución) cada vez que se haga una petición se escribirá la hora, la cadena y el número.

```
String timeStamp = new SimpleDateFormat("yyyy.MM.dd HH.mm.ss").format(new Date());
```

Puntos:

- 1 punto. Multithread HTTP.
- 2 punto. Generar y mandar el HTML al cliente.
- 2 punto. Se guarda el log en una ruta relativa.

UDP

Crea un programa que escuche en un puerto recibido. El programa espera recibir números. Implementará el patrón observer.

Habrán tres observadores esperando números. Uno de ellos cuando recibe el número genera una pirámide y la escribe por pantalla. Otro genera un rombo y lo escribe por pantalla. El otro escribe si el número es o no primo.

El programa se cierra si recibe un `-1`.

Puntos:

- 1 punto. Escuchador con observer.
- 1 punto. Suscripción y pintado de pirámide
- 1 punto. Suscripción y pintado de rombo
- 1 punto. Suscripción y pintado de primo-noprime
- 1 punto. Termina con `-1`