

## EJERCICIO 1

Este script utiliza un bucle for para iterar sobre una secuencia de números del 1 al 200. La secuencia {1..200..2} genera todos los números impares entre 1 y 200, ya que el tercer argumento en la secuencia (2) indica el incremento entre cada número. En cada iteración, el script imprime el número actual.

## EJERCICIO 2

Primero, este script crea dos archivos llamados LRSO1.txt y LRSO2.txt utilizando el comando touch, que crea archivos vacíos si no existen. Luego, utiliza un bucle for para iterar sobre todos los archivos en el directorio actual que coinciden con el patrón LRSO\*.txt. El asterisco (\*) en el patrón es un carácter comodín que coincide con cualquier cadena de caracteres. Por lo tanto, LRSO\*.txt coincide con cualquier archivo que comienza con LRSO y termina con .txt. En cada iteración, el script imprime el nombre del archivo actual.

## EJERCICIO 3

Este script recibe dos parámetros numéricos de entrada y devuelve la resta del mayor número respecto al menor. Si no se proporcionan suficientes parámetros, el script solicita los valores faltantes al usuario utilizando el comando read, que lee la entrada del usuario desde la línea de comandos. Luego, compara los dos números utilizando el operador -gt, que devuelve verdadero si el primer número es mayor que el segundo. Dependiendo del resultado de la comparación, el script realiza la resta correspondiente utilizando el operador \$(), que permite realizar operaciones aritméticas en bash.

## EJERCICIO 4

Este script es una calculadora en bucle infinito que ofrece dos operaciones: potenciación y conversión de grados Kelvin a Celsius. El bucle while true crea un bucle infinito que solo se detiene cuando el script se interrumpe manualmente. Dentro del bucle, el script presenta un menú al usuario y lee la opción seleccionada con el comando read. Dependiendo de la opción seleccionada, el script realiza la operación correspondiente. Para la potenciación, se utiliza un bucle for para multiplicar la base por sí misma un número de veces igual al exponente. Para la conversión de temperatura, se utiliza la fórmula de conversión proporcionada.

## EJERCICIO 5

Este script recibe un parámetro de entrada, que debe ser un nombre de archivo. El script verifica si el archivo existe en el directorio actual utilizando el operador -f, que devuelve verdadero si el archivo existe y es un archivo regular. Si el archivo existe, se imprime un mensaje. Si no existe, se crea el archivo y se escribe una línea de texto en él utilizando el operador de redirección >, que redirige la salida de un comando a un archivo.

## EJERCICIO 6

Este script recibe un parámetro de entrada que determina el formato en el que se mostrará la fecha actual. El script verifica el número de parámetros proporcionados y muestra un mensaje de error si no es correcto. Luego, utiliza una sentencia case para determinar qué acción realizar en función del parámetro proporcionado. Si se pasa el parámetro 'c' o 'corta', se muestra la fecha en el formato 'DIA/MES/AÑO' utilizando el comando date con el formato apropiado. Si se pasa el parámetro 'l' o 'larga', se muestra un mensaje más detallado con la fecha. Si no se pasa ningún parámetro, se muestra el calendario del mes actual utilizando el comando cal.