

# Programación

---

## Ejercicio 1: Cifrado Atbash (Listas o ArrayList)

Implementar un método que calcule el método de cifrado en el oriente medio denominado “Atbash”.

El cifrado Atbash es un simple cifrado por sustitución que se basa en la transposición de todas las letras del alfabeto de forma que el alfabeto resultante esté al revés. La primera letra se sustituye por la última, la segunda por la penúltima, y así sucesivamente.

Un cifrado Atbash para el alfabeto latino sería el siguiente:

Original: abcdefghijklmnopqrstuvwxyz

Cifrado: zyxwvutsrqponmlkjihgfedcba

Es un cifrado muy débil porque sólo tiene una clave posible, y es un simple cifrado de sustitución monoalfabética. Sin embargo, esto puede no haber sido un problema en la época del cifrado.

### Ejemplos:

Atbash(“test”) -> “gvhg”

Atbash(“gsvjf rxpyi ldmul cqfnk hlevi gsvoz abwlt”) ->  
“thequickbrownfoxjumpsoverthelazydog”

**Hint:** Mostrar el proceso en la consola del sistema

## Ejercicio 2: Notación posicional (Estructura a libre elección)

Implementar un método ToAnyBase que tomé por parámetro un número entero y una base  $n$  donde  $n \geq 2$ . La implementación de la función deberá permitir que logre convertir el número, representado como una secuencia de dígitos en la base recibida en el segundo parámetro.

## Notación posicional

En notación posicional, un número en base  $b$  puede entenderse como una combinación lineal de potencias de  $b$ .

### Ejemplo de uso:

El número 42, en base 10, significa

$$(4 \cdot 10^1) + (2 \cdot 10^0)$$

El número 101010, en base 2, significa:

$$(1 \cdot 2^5) + (0 \cdot 2^4) + (1 \cdot 2^3) + (0 \cdot 2^2) + (1 \cdot 2^1) + (0 \cdot 2^0)$$

El número 1120, en base 3, significa

$$(1 \cdot 3^3) + (1 \cdot 3^2) + (2 \cdot 3^1) + (0 \cdot 3^0)$$

**Hint:** Mostrar el proceso en la consola del sistema

## Ejercicio 3: Producto contiguo (Pilas)

Implementar un método `seriesProduct` que reciba por parámetro una cadena de dígitos y un número  $n$  que represente el tamaño de la subcadena que se desea encontrar. La subcadena contigua de longitud  $n$  dígitos buscada debe obtener el producto más grande posible. Finalmente la función deberá retornar el valor obtenido del producto más grande posible de la subcadena de longitud  $n$ .

### Ejemplo de uso:

`seriesProduct("1027839564",3)` -> 270 (obtenido de calcular  $9 \cdot 5 \cdot 6$ )

**Hint:** El usuario solo podrá cargar valores enteros en caso contrario se debe avisar del error y reiterar la solicitud del valor en cuestión.

## Ejercicio 4: Dragones y mazmorras (Estructura a libre elección)

En una partida de Dragones y Mazmorras, cada jugador comienza generando un personaje con el que puede jugar. Este personaje tiene, entre otras cosas, seis habilidades: fuerza, destreza, constitución, inteligencia, sabiduría y carisma. Estas seis habilidades tienen puntuaciones que se determinan al azar. Para ello, se tiran cuatro dados de 6 caras y se anota la suma de los tres dados más grandes. Lo haces seis veces, una por cada habilidad.

Los puntos de golpe iniciales de tu personaje son  $10 +$  el modificador de constitución de tu personaje.

El modificador de constitución de tu personaje se obtiene restando 10 a la constitución de tu personaje, dividiendo por 2 y redondeando hacia abajo. Se solicita implementar un método `GameGenerator` que genere personajes aleatorios que sigan las reglas anteriores.

Por ejemplo, los seis lanzamientos de cuatro dados pueden ser como:

- 5, 3, 1, 6: Descartas el 1 y sumamos  $5 + 3 + 6 = 14$ , que asignas a la fuerza.
- 3, 2, 5, 3: Descartas el 2 y sumamos  $3 + 5 + 3 = 11$ , que asignas a la destreza.
- 1, 1, 1, 1: Descartas el 1 y sumamos  $1 + 1 + 1 = 3$ , que asignas a constitución.
- 2, 1, 6, 6: Descartas el 1 y sumamos  $2 + 6 + 6 = 14$ , que asignas a inteligencia.
- 3, 5, 3, 4: Descartas el 3 y sumamos  $5 + 3 + 4 = 12$ , que asignas a la sabiduría.
- 6, 6, 6, 6: Descartas el 6 y sumamos  $6 + 6 + 6 = 18$ , que asignas a carisma.

Como la constitución es 3, el modificador de constitución es  $-4$  y los puntos de golpe son 6.