

## PROGRAMACIÓN

### PRUEBA OBJETIVA DICIEMBRE

<b>SE VALORARÁ POSITIVAMENTE:</b>
1. Los algoritmos y las estructuras de datos utilizados para la solución así como su justificación.
<b>SE PENALIZARÁ:</b>
2. Los errores en los requisitos y los algoritmos, teniendo en cuenta que un error leve descuenta 1 punto y un error grave 2 puntos. <ul style="list-style-type: none"><li>• Se considera un error leve aquel, que pese a su existencia, permite una solución al problema.</li><li>• Un error grave es aquel que imposibilita una correcta y exacta solución al problema.</li><li>• La no utilización del método, de cada ejercicio, en un <b>programa principal será un leve</b>.</li></ul>
Los ejercicios valen 2,5 puntos y se debe respetar el tiempo máximo establecido y las normas de permanencia. Recomendable utilizar, como método de resolución, el visto en clase pero se corregirá sobre Java.
<b>EVALUACIÓN DE LOS RESULTADOS DE APRENDIZAJE: RA1, RA2, RA3 Y RA4</b>

**IMPORTANTE: DEBÉIS UTILIZAR CADA MÉTODO EN UN PROGRAMA PRINCIPAL.**

**Nombre y apellidos:**

**12/12/2024**

**Firma:**

1. La agencia secreta "Los Guardianes del Código" necesita vuestra ayuda para identificar agentes infiltrados. Los agentes de la agencia poseen un identificador único con un formato especial: ocho dígitos seguidos de una letra mayúscula (excepto la Ñ).

Implementa en Java el método `esAgenteSecreto(char[] identificador)` que recibe un array de caracteres identificador y determina mediante un booleano si cumple con el formato de los agentes de "Los Guardianes del Código". Si el identificador es válido, devuelve `true`; de lo contrario, devuelve `false`.

Ejemplos de uso:

`esAgenteSecreto({'0','3','4','4','5','0','8','4','Y'});` → `true`  
`esAgenteSecreto({'2','3','4',';','*','5','0','8','5','V'});` → `false`

Puedes usar las siguientes funciones auxiliares:

```
static boolean esLetraMayusculaMenosEnye (char caracter) {  
    return 'A' <= caracter && caracter <= 'Z' && caracter != 'Ñ';  
}  
static boolean esDigitoValido (char caracter) {  
    return '0' <= caracter && caracter <= '9';  
}
```

## PROGRAMACIÓN

2. Los duendes del bosque encantado se comunican mediante un lenguaje peculiar: ¡hablan al revés! Para entender sus mensajes secretos, necesitamos un traductor especial.

Implementa en Java el método `traductorDuende(String mensaje)` que recibe un `String` mensaje en lenguaje humano y devuelve un `String` que contiene la traducción al idioma de los duendes, invirtiendo el orden de las letras.

Ejemplos de uso:

`traductorDuende("seta");` → `"ates"`

`traductorDuende("bosque");` → `"euqsob"`

Se puede usar la siguiente función:

```
static int aDigito (char character) {  
    return character - '0';  
}
```

3. El gran mago Merlín guarda sus ingredientes mágicos (representados por números enteros) en un cofre encantado. Para mantener el orden, los ingredientes están organizados en una secuencia numérica ascendente dentro del cofre.

Implementa en Java el método `tieneIngrediente(int[] cofre, int ingrediente)` que recibe un array de enteros `cofre` (que representa la secuencia ordenada de ingredientes) y un entero `ingrediente`. El método debe determinar si el ingrediente mágico especificado existe dentro del cofre. Si el ingrediente se encuentra, devuelve `false`; de lo contrario, devuelve `true`.

Ejemplos de uso:

`int[] cofre = {2, 5, 8, 12, 16, 23, 38, 56, 72, 91};`

`tieneIngrediente(cofre, 23);` → `false`

`tieneIngrediente(cofre, 15);` → `true`

## PROGRAMACIÓN

4. ¡El famoso parque de atracciones Hulkland, propiedad del mismísimo Dr. Bruce Banner, tiene una oferta especial! El precio de la entrada depende del número de veces que hayas visitado el parque. Las dos primeras visitas cuestan 200 smashcoins cada una (la moneda oficial de Hulkland). Si vienes por tercera, cuarta o quinta vez, el precio baja a 150 smashcoins por entrada. Y si eres un auténtico fan y nos visitas 6 veces o más, ¡cada entrada te costará solo 100 smashcoins!

Implementa en Java el método `precioEntradaHulkland(int numVisitas)` que recibe el número de veces que un cliente ha visitado Hulkland (`numVisitas`) y devuelve el precio de su próxima entrada en smashcoins.

Ejemplos de uso:

*`precioEntradaHulkland(1);` → 200*  
*`precioEntradaHulkland(3);` → 150*  
*`precioEntradaHulkland(7);` → 100*