

# EXAMEN MODELO 2





Nombre y apellidos: \_\_\_\_\_

Ejercicio	1	2	3	Total
Puntos	3	3	4	10
Nota				

1. (3 puntos) Completa las especificaciones o definiciones de los siguientes subprogramas que se describen a continuación. Si no necesitas alguno de los elementos (precondiciones, postcondiciones, parámetros, ...), tacha el espacio que se ha dejado para completar la documentación del subprograma.

- (a) La función *contarCuatrosEnNumero*, dado un entero positivo, devuelve cuántos 4 hay entre los dígitos de dicho número.

```
/**
 * Función que permite obtener el número de 4s entre
 * los dígitos de un número.
 * <p><strong>PRE:</strong>
 *
 * </p>
 * <p><strong>POST:</strong>
 *
 * </p>
 *
 * @param
 *
 * @return
 */
public static int contarCuatrosEnNumero (int pNum)
```

- (b) La función **posicionEnCadena**, dadas dos cadenas de caracteres *pCad1* y *pCad2*, devuelve en qué posición está *pCad2* dentro de *pCad1*. Si *pCad2* no está incluida en *pCad1*, devuelve -1.

```
/**
 * Función que permite obtener la posición en la que se
 * encuentra pCad2 dentro de pCad1
 * <p><strong>PRE:</strong>
 *
 * </p>
```



```
* <p><strong>POST:</strong>
*
* </p>
*
* @param
*
*
*
* @return
*/
public static int posicionEnCadena (String pCad1,
                                   int pCad2)
```

- (c) La función **ponerSubcadenaEnMayusculas**, dados una cadena de caracteres, un entero positivo que indica la posición inicial de la subcadena y un entero positiva que indica el final de la subcadena, devuelve la cadena que se obtiene al pasar la subcadena a mayúsculas.

```
/**
 * Función que devuelve la cadena de caracteres que se
 * obtiene al pasar la subcadena delimitada por
 * pPos1 y pPos2 a mayúsculas.
 *
 * <p><strong>PRE:</strong>
 * pPos1 >= 0 y pPos1 < numLetras(pCadena) y
 * pPos1 >= 0 y pPos2 < numLetras(pCadena) y pPos2 >pPos1
 * </p>
 *
 * @param pCadena - la cadena que se quiere convertir
 * @param pPos1 - la posición inicial de la subcadena
 * a pasar a maúsculas
 * @param pPos2 - la posición final de la subcadena
 * a pasar a maúsculas
 * @return la cadena resultante
 */
public static _____ ponerSubcadenaEnMayusculas
                      (_____)
```

2. (3 puntos) **Diseña** un subprograma que dada una cadena de caracteres que representa un DNI (supón que la cadena tiene el formato adecuado), devuelve *true* si es válido



(la letra corresponde al resto obtenido al dividir la suma de los dígitos del número del DNI por 23<sup>1</sup>) y *false* en caso contrario. **Debes utilizar los subprogramas que se especifican a continuación:**

**obtNumDNI**

**Param:**

pDNI: Cadena

**PRE:** la cadena pDNI contiene un DNI con formato adecuado

**POST:** devuelve el número del DNI

**Devuelve** Entero

**obtLetraDNI**

**Param:**

pDNI: Cadena

**PRE:** la cadena pDNI contiene un DNI válido

**POST:** devuelve la letra del DNI

**Devuelve** Character

**obtLetraValor**

**Param:**

pValor: Entero

**Devuelve** Character

**PRE:**  $pValor \geq 0$  y  $pValor \leq 22$

**POST:** devuelve la letra de DNI que le corresponde al valor pValor

**obtNumDigitos**

**Param:**

pNum: Entero

**POST:** devuelve el número de dígitos de pNum

**Devuelve** Entero

**obtDigitoPos**

**Param:**

pNum: Entero

pPos: Entero

**PRE:**  $pNum \geq 0$  y  $pPos \geq 0$  y  $pPos < obtNumDigitos(pNum)$

**POST:** devuelve el dígito de la posición pPos de pNum

**Devuelve** Entero

Si necesitas algún subprograma adicional, incluye también su **diseño y especificación** en la respuesta al ejercicio.

---

<sup>1</sup>Este no es el método real



3. (4 puntos) **Implementa** un subprograma que lee una secuencia de números acabada en 0, y muestra para cuántos de esos números se obtiene un valor par al sumar sus dígitos. Por ejemplo, si la secuencia de números es

1   24   34   0

el subprograma debe mostrar 1. **Implementa** también todos los subprogramas que necesites para este ejercicio.