

# EXAMEN MODELO 1





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 *contarVocalesFrase*, dada una cadena de caracteres, devuelve el número de vocales que contiene dicha cadena.

```
/**
 * Función que permite obtener el número de vocales de la
 * cadena de caracteres proporcionada.
 * <p><strong>PRE:</strong>
 *
 * </p>
 * <p><strong>POST:</strong>
 *
 * </p>
 *
 * @param
 *
 * @return
 */
public static int contarVocalesFrase (String pFrase)
```

- (b) La función **maximoComunDivisor**, dados dos números enteros positivos tales que  $pNum1 \geq pNum2$ , devuelve el máximo común divisor de dichos números.

```
/**
 * Función que permite obtener el máximo común divisor de
 * dos números enteros positivos.
 * <p><strong>PRE:</strong>
 *
 * </p>
 * <p><strong>POST:</strong>
```



```
*  
* </p>  
*  
* @param  
*  
*  
*  
* @return  
*/  
public static int maximoComunDivisor (int pNum1,  
                                     int pNum2)
```

- (c) La función **cambiarBase**, dados un número entero mayor o igual que cero, y un entero positivo que representa la base, devuelve el valor correspondiente al número dado representado en la base indicada.

```
/**  
 * Función que devuelve el valor de pNum en base pBase.  
 *  
 * <p><strong>PRE:</strong></p>  
 * pNum >= 0 y pBase >= 1  
 * </p>  
 *  
 * @param pNum - el valor que se quiere convertir  
 * @param pBase - la base en la que se quiere representar  
 * el valor pNum  
 * @return el valor de pNum en base pBase  
 */  
public static ____ cambiarBase (____)
```

2. (3 puntos) **Diseña** un subprograma que lea una secuencia de cadenas de caracteres acabada por la cadena “@fin” y muestre en la salida estándar cuántas de las cadenas de caracteres tienen un número par de vocales. Por ejemplo, dada la secuencia de cadenas de caracteres

“Hola” “Fin” “Ahora” “@fin”

el subprograma debe mostrar 1, ya que “Hola” es la única cadena con un número par de vocales.

**Debes utilizar los subprogramas que se especifican a continuación:**



**leeCadena**

**POST:** devuelve la cadena de caracteres introducida por el usuario

**Devuelve** Cadena

**obtNumLetras**

**Param:**

pCadena: Cadena

**POST:** devuelve el número de letras de pCadena

**Devuelve** Entero

**obtLetraPos**

**Param:**

pCadena: Cadena

pPos: Entero

**Devuelve** Character

**PRE:**  $pPos \geq 0$  y  $pPos < obtNumLetras(pCadena)$

**POST:** devuelve la letra en la posición pPos de pCadena, siendo 0 la posición de la primera letra.

**esVocal**

**Param:**

pLetra: Character

**POST:** devuelve *true* si el caracter es vocal y *false* en caso contrario

**Devuelve** booleano

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

3. (4 puntos) **Implementa** un subprograma que lee una secuencia de números acabada en 0, y muestra cuántos tienen dígitos con valor impar. Por ejemplo, si la secuencia de números es

1 24 34 0

el subprograma debe mostrar 2. **Implementa** también todos los subprogramas que necesites para este ejercicio.