

## Subprogramas y Refinamientos sucesivos.

### 1. Escribir la especificación de las siguientes funciones (cabecera y pre/pos):

- 1.1 Obtenga el valor mínimo de dos números enteros.
- 1.2 Determine si un número natural es primo (primo(1) y no primo(0))
- 1.3 Determine si un número positivo es capicúa.
- 1.4 Diga si una letra es minúscula.
- 1.5 Diga si un número natural es múltiplo de 3.
- 1.6 Diga si dos números enteros están ordenados.

### 2. Escribir la especificación de los siguientes procedimientos (cabecera y pre/pos):

- 2.1 Diga si un número natural acaba en 3.
- 2.2 Dado un número entero positivo, diga si es perfecto (un número entero positivo es perfecto si es igual a la suma de sus divisores (sin incluirse el mismo), p. ej.:  $28 = 1 + 2 + 4 + 7 + 14$ ).
- 2.3 Dado un string de caracteres y un carácter, el subprograma devuelve el carácter del string que sigue al caracter de entrada. Si el carácter es el último del string, o no está, se devuelve el mismo carácter de entrada. Este subprograma también devuelve un valor booleano que indica si se encontró el carácter en el string.

### 3. Especificar los siguientes programas (cabecera y pre/pos) y diseñar el algoritmo correspondiente a cada uno de ellos teniendo en cuenta los subprogramas definidos en los ejercicios 1 y 2.

- 3.1 Dado un número N natural en la entrada estándar escribir los N primeros números naturales que no sean múltiplos de 3 ni acaben en 3.
- 3.2 Dado un número N natural escriba todos los números positivos menores que N que sean perfectos.

### 4. Sea la declaración de los siguientes subprogramas:

```
public static int min(int X, int Y, int M);
```

```
/**
```

```
 * POS: el resultado es el mínimo de los dos valores
```

```
 **/
```

```
public static int max(int X, int Y);
```

```
/**
```

```
 * POS: el resultado es el máximo de los dos valores
```

```
 **/
```

```
public static void actualizar (int total, boolean condicion, int añadir);
```

```
/**
```

```
* POS: si la condición es cierta, devuelve resultado de incrementar el valor total tras añadir
* sumarle la cantidad añadir
**/
```

Indica cuál de estos usos de los subprogramas anteriores son incorrectos, teniendo en cuenta que: *num1, num2,resul: int; y exento:boolean;*

a) num1=3; num2=6;  
resul=max(num1,num2);

b) num1=3; num2=6;  
resul=min(num1,num2);

c) num1=3; num2=6;  
if (max(num1,num2) >10)  
{ min(num1,num2,resul);  
  System.out.println(min(num1,num2,resul));  
  System.out.println(resul);  
  System.out.println(max(num1,num2));  
}

d) num1=3; num2=6;  
System.out.println(min(num1,num2,resul));

e) num1 = 10.000; exento = false;  
resul =actualizar(num1, ! exento, 2500)

f) num1 = 10.000;  
actualizar(num1 < 500.000, num1, 2.500);

g) num1 = 10.000; exento = false;  
num1 = actualizar(!exento, 2500);

h) num1=14;  
result =actualizar(14, true, 98);

i) num1=14;  
actualizar(num1, true, 98);

j) num1=3; num2=8; exento=false;  
resul=actualizar(num1,exento,num2);

## 5. Implementar en Java los siguientes subprogramas:

**5.1** Dado un número entero positivo, diga si es perfecto (un número entero positivo es perfecto si es igual a la suma de sus divisores (sin incluirse el mismo), p. ej.:  $28 = 1 + 2 + 4 + 7 + 14$ ).

**5.2** Dado un número N entero positivo en la entrada estándar obtiene el primer número menor que N que sea primo y perfecto. Si no lo encuentra devuelve el mismo N.

También devuelve un valor booleano dependiendo de si dicho número primo y perfecto existe.

**6. Especificar** los siguientes problemas (cabecera y pre/pos) y **diseñar el algoritmo** correspondiente teniendo en cuenta los subprogramas definidos en los ejercicios 1, 2 y en el ejemplo *calendario* (Teoría)

**6.1** En la entrada estándar viene una secuencia de números terminada en cero. El **programa** escribe en la salida estándar el mínimo de todos ellos (sin incluir el cero).

**6.2** En la entrada estándar viene una secuencia de números. El **programa** escribe en la salida estándar los que son primos.

**6.3** En la entrada estándar tenemos una secuencia de caracteres terminada en punto. El programa devuelve el número de vocales en dicha secuencia.

**6.4** En la entrada estándar viene una secuencia de números. El **programa** devuelve cuántos de ellos son perfectos y además los escribe en la salida estándar.

**6.5** En la entrada estándar viene una secuencia de pares (string y carácter). El **subprograma devuelve en la salida estándar** los caracteres que no aparecen en su string.

**6.6 .** Dado un día de los siglos XX ó XXI, el **subprograma devuelve** el día siguiente (tener en cuenta que el 1 de enero de 1900 fue lunes).