

# Introducción a la Computación

Grado en Ingeniería Química, curso 1<sup>ero</sup>. FCT/ZTF UPV/EHU

Examen con ordenador. Scilab.

20 de diciembre de 2016

Con soluciones.

(otras pueden ser igualmente válidas)

---

Sin apuntes. Teléfonos apagados o silenciados y a la vista.

1 hora desde el momento de entrega de este enunciado

Subir el resultado a egela (link "EXAMEN 20/12/2016"). Estará abierto los últimos 5 minutos. No se puede salir hasta el final.

---

**Ejercicio1.-** Escribir una función que recibe como parámetro una variable vector de números y lo devuelve después de aplicar a todos ellos una función consistente en doblar su valor y saturar el resultado de modo que si pasa de 0.75 sea 0.75 y si no llega a 0.25 sea 0.25.

El programa principal se limitará a comprobar el funcionamiento generando un pequeño vector con valores aleatorios [ `x=rand(10,1)` ] y mostrando su valor antes de aplicarle la función y después. (puede utilizar la función `disp()` de scilab)

```
function v=ejercicio1(v)
    for i=1:length(v)
        v(i)=min(0.75, max(0.25,v(i)*2))
    end
endfunction

//Se prueba como se indica en el enunciado
x=rand(10,1)
disp(x)
disp(ejercicio1(x))
```

---

**Ejercicio2.-** Basándonos en la definición de "números amigos" que se vio en las prácticas, podemos definir un número "perfecto" como "el que es amigo de sí mismo". Es decir, un número perfecto es aquel que coincide con la suma de sus divisores propios. Un ejemplo es el siguiente:

divisores de 6 = {1,2,3};  $1+2+3=6 \Rightarrow$  el 6 es un número "perfecto"

Escribir la función "esPerfecto" que recibirá como parámetro un número y devolverá un resultado booleano en función de que sea perfecto o no (%T si es perfecto, %F si no lo es)

El programa principal se limitará a pedir un número al usuario y decirle si es perfecto o no usando la función anterior.

(comprobar que se ha resuelto correctamente haciendo pruebas sabiendo que 1, 6, 28, 496, 8128 son los únicos números perfectos por debajo de 1.000.000)

```
function b=esPerfecto(n)
    suma=0
    for i=1:n-1
        if (modulo(n,i)==0) then suma=suma+i; end;
    end
    b= n==suma
endfunction

//El programa principal pedido
p=input("introduce un número para comprobar s es perfecto: ")
if esPerfecto(p) then disp("Es perfecto"); else disp("No es perfecto"); end
```

### Variantes del ejercicio1...

La mostrada junto al enunciado es una solución bastante "elegante". No obstante es posible que sólo los que hayan practicado la programación con "verdadero interés" hayan sabido usar las funciones máximo [max(a,b)] y mínimo [min(a,b)] . Una solución más al alcance de cualquiera es esta:

```
function v=ejercicio1(v)
    for i=1:length(v)
        v(i)=2*v(i)
        if v(i)>0.75 then v(i)=0.75; end
        if v(i)<0.25 then v(i)=0.25; end
    end
endfunction
```

Por otro lado, el enunciado dice "aplicar a todos ellos una función consistente en doblar...", y aunque no era la intención que se tratara de una función de Scilab (sino meramente una transformación del valor), sí que induce a interpretarlo de ese modo. En todo caso, no añade dificultad... quedaría así (con la solución "elegante")

```
function d=transformaNumero(d)
    d=min(0.75, max(0.25,2*d))
endfunction

function v=ejercicio1(v)
    for i=1:length(v)
        v(i)=transformaNumero(v(i))
    end
endfunction
```