

# Práctica 5

## Resolución de problemas de selección

Grado en Ingeniería en Sistemas Audiovisuales y Multimedia

GSyC, Universidad Rey Juan Carlos

6 de octubre de 2017

## 1. Introducción

Seguimos viendo formas de resolver problemas. La pasada semana ya vimos y estuvimos haciendo ejercicios cuyas soluciones sólo requerían de la aplicación de una fórmula; eran problemas de resolución directa.

Esta semana ya ha tocado el turno a los problemas de selección, cuyo símil hemos visto que son las funciones definidas a trozos. Este tipo de problemas no tiene un algoritmo único que lo resuelva, sino que depende del valor que se le dé de entrada. De igual forma que una función a trozos no tiene la misma definición en todo su dominio, sino que depende del valor sobre el que está siendo evaluada/definida.

Así, en la práctica de hoy vamos a intentar resolver varios problemas algo más elaborados que los previos, en los que pondremos énfasis en el uso de *estructuras de control* (*if-else*) y *bucles*.

**Recuerda que es importante resolver los problemas por ti mismo. En clase y, sobretodo, en la Web hay muchos recursos que te servirán de ayuda; pero éstos nunca podrán pensar por ti...**

## 2. Problemas a resolver

### 2.1. Figuras de las cartas

Devolver un valor de verdad (`bool`) que indique si un número de carta (de 1 a 10) corresponde a una figura y, en su caso, qué figura es (`sota`, `caballo` o `rey`).

### 2.2. Juego de las 7 y media

Devolver el valor de una carta en el juego de las 7 y media. En este juego, a efectos de sumar los deseados siete puntos y medio de cada jugador, cada carta tiene, independientemente de su palo, el valor que indica su propio índice, salvo las doce figuras (sotas, caballos y reyes) que tienen un valor de medio punto.

### 2.3. Ecuación de primer grado

Escribe un programa que reciba los coeficientes `a` y `b` de la ecuación de primer grado  $ax + b = 0$  y que muestre por pantalla una de las tres posibilidades: `Tiene solución única`, `Tiene infinitas soluciones`, o `No tiene solución`.

### 2.4. Signo del zodiaco

Determinar el signo zodiacal de una persona. Éste está determinado por el día de nacimiento según las siguientes tuplas `Desde (mes, día)`, `Hasta (mes,día)`.

```

'aries':      (( 3, 21), ( 4, 20)),
'tauro':     (( 4, 21), ( 5, 21)),
'geminis':   (( 5, 22), ( 6, 21)),
'cancer':    (( 6, 22), ( 7, 23)),
'leo':       (( 7, 24), ( 8, 23)),
'virgo':     (( 8, 24), ( 9, 23)),
'libra':     (( 9, 24), (10, 23)),
'escorpio':  ((10, 24), (11, 22)),
'sagitario': ((11, 23), (12, 21)),
'capricornio': ((12, 22), ( 1, 20)),
'acuuario':  (( 1, 21), ( 2, 19)),
'piscis':    (( 2, 20), ( 3, 20)),

```

Por ejemplo, para que una persona sea de signo **libra** debe haber nacido entre el 24 de septiembre y el 23 de octubre.

Escribe la función `determinarSigno(int dia, int mes)` que reciba como parámetro la fecha de nacimiento de una persona (`día`, `mes`), y que devuelva el valor del signo zodiacal de la persona; esto es, 1 para **aries**, 2, para **tauro**, ... hasta el valor 12 para **piscis**.

Este valor lo emplearás como parámetro de entrada de un procedimiento `mostrarSigno (int valor)`, cuyo cometido sea mostrar por pantalla el signo zodiacal correspondiente a tal valor de entrada.

### 3. Entrega

Deberás crear un nuevo fichero por cada problema a resolver, con nombre **apellido-nombre-pX.p**, donde **X** será el número del problema que resuelve.

La entrega de esta práctica se hará de forma no presencial, dejando todos los ficheros generados contenido en un fichero **apellido-nombre-P05.tar** que habrá que dejar adjunto en la tarea (Moodle).

Fecha de entrega: **12 de octubre**

Se podrá entregar hasta las 23:59h de este día.