

---

# Práctica Profesionalizante

## Herramientas de Programación

### Funciones

1. Desarrollar una función que reciba tres números enteros positivos y devuelva el mayor de los tres, sólo si éste es único (es decir el mayor estricto). Devolver **None** en caso de no haber ninguno. Desarrollar también un programa de prueba para ingresar los tres valores, invocar a la función y mostrar el máximo hallado, o un mensaje informativo si éste no existe.
2. Hacer una función que nos indique con verdadero o falso si una fecha ingresada en el formato día, mes, año es válida o inválida. Considerar que será conveniente hacer una función que verifique si un año es bisiesto, considerando que un año es bisiesto si es divisible por 4 pero no es divisible por 100, o en el caso que sea divisible por 400. Desarrollar el código para probarla.
3. Realizar un programa que pida al usuario que ingrese el día y mes de su cumpleaños (aplique la función de validación de fecha del punto anterior) y el programa le debe decir a qué [signo del zodiaco](#) corresponde.  

<b>Aries:</b> 21 de marzo al 20 de abril	<b>Tauro:</b> 21 de abril al 20 de mayo
<b>Géminis:</b> 21 de mayo al 21 de junio	<b>Cáncer:</b> 22 de junio al 23 de julio
<b>Leo:</b> 24 de julio al 23 de agosto	<b>Virgo:</b> 24 de agosto al 23 de septiembre
<b>Libra:</b> 24 de septiembre al 22 de octubre	<b>Escorpio:</b> 23 de octubre al 22 de noviembre
<b>Sagitario:</b> 23 de noviembre al 21 de diciembre	<b>Capricornio:</b> 22 de diciembre al 20 de enero
<b>Acuario:</b> 21 de enero al 19 de febrero	<b>Piscis:</b> 20 de febrero al 20 de marzo
4. Una empresa necesita un programa para la tesorería que le proporcione al cajero el cambio exacto que debe entregarle al solicitante par pagar una serie de importes. Para eso se necesita desarrollar una función que reciba 2 listas, la primera con todos los importes que hay que abonar, la segunda con los valores de los billetes disponibles. Por ejemplo si hay que abonar [ 1320, 16350, 23280 ] y los importes disponibles son billetes [5000, 1000, 500, 200, 100, 50, 10].

### Funciones recursivas

5. Escribir una función que devuelva la cantidad de dígitos de un número entero, sin utilizar cadenas de caracteres.
6. Desarrollar una función que devuelva el producto de dos números enteros por sumas sucesivas.

- 7.** Se definen los números triangulares  $T_N$  como los obtenidos de sumar todos los números naturales sucesivos  $1 + 2 + 3 + \dots + N$ . Por ejemplo  $T_1 = 1$ ,  $T_2 = 3$ ,  $T_3 = 6$ ... etc. Realizar una función recursiva que muestre la sucesión de números triangulares desde  $T_1$  hasta  $T_N$
- 8.** Desarrollar una función que reciba un número en base decimal y lo devuelva transformado en base binaria.
- 9.** Ingrese el numerador y el denominador de un número fraccionario, devolver la fracción simplificada utilizando el algoritmo de euclides para encontrar el máximo común divisor. El algoritmo de euclides en forma recursiva es muy simple: siendo  $MCD(v1, v2)$ :
  - si  $v1$  es igual a  $v2$  entonces el MCD es  $v1$  (o  $v2$ ),
  - si  $v1$  es mayor a  $v2$  entonces es el  $MCD(v1-v2, v2)$
  - si  $v2$  es mayor a  $v1$  entonces es el  $MCD(v1, v2-v1)$

## Módulos

- 10.** Crear un programa que simule el funcionamiento de una máquina tragamonedas con un crédito inicial de 1000 fichas. El funcionamiento básico del tragamonedas es el siguiente:
  - Inicialmente el jugador indica cuánto de las fichas disponibles quiere jugar (apuesta) y se le resta esa cantidad de las fichas.
  - El programa genera tres números al azar del 1 al 5 y los muestra en pantalla.
  - Si los tres números son distintos, el jugador no acumula nada por lo tanto perdió el valor apostado.
  - Si salen dos números iguales, el jugador acumula el doble de lo apostado.
  - Si salen tres números iguales, el jugador acumula lo apostado multiplicado por cinco.
  - Si todavía le quedan fichas, el jugador indica si quiere seguir jugando o se retira.
- 11.** Crear una función que devuelva una lista que corresponda a un mazo de cartas españolas (del 1 al 12) sin comodines, la representación de cada carta será una tupla (palo, valor) por ejemplo el as de bastos se representa ('bastos', 1). Luego crear una función que mezcle el mazo al azar con la siguiente metodología: por cada carta del mazo intercambiarla por cualquiera al azar que se encuentre en cualquier posición. Devolver el mazo mezclado. Crear el código para demostrar la funcionalidad.