

Programación Orientada a Objetos

Taller 1: Repaso de Programación Estructurada

Para cada uno de los siguientes problemas, escriba **una función** que lo solucione. La función debe tener los parámetros indicados en el enunciado.

Además, haga un programa (función *main*) que invoque cada una de las funciones, tantas veces como crea necesario, para verificar su correcto funcionamiento.

Recuerde que un factor determinante es la simplicidad de sus soluciones.

Problema 1

Dado un entero $n \geq 0$, determinar su representación en sistema binario (base 2).

La función recibe n como parámetro y debe retornar una cadena (string) de ceros y unos que sea la representación en binario de n .

```
def decimalToBin(n):
```

Problema 2

En un museo se tiene la siguiente tabla de tarifas para cobrar el valor del ingreso. La tarifa depende la edad del visitante.

Edad	Valor de la entrada
Hasta 2 años	0
Mayores de 2 años y menores de 13	14000
Desde 13 hasta 61 años	25000
De 62 años en adelante	18000

Dadas las edades de un grupo de visitantes, se debe calcular el costo total de las entradas.

La función recibe como parámetro un arreglo que guarda las edades de los visitantes y debe retornar el costo total de las entradas para ese grupo de personas.

```
def ticketsCost(ages):
```

NOTA: Para crear arreglos de Python, primero deberá importar el módulo **array** que contiene todas las funciones necesarias.

Problema 3

El zodíaco chino asigna animales a los años, en un ciclo de 12 años. En la siguiente tabla se muestra un ciclo de 12 años. El patrón se repite a partir de ahí, siendo 2012 otro año del dragón y 1999 otro año de la liebre.

Año	Animal
2000	Dragón
2001	Serpiente
2002	Caballo
2003	Oveja
2004	Mono
2005	Gallo
2006	Perro

2007	Cerdo
2008	Rata
2009	Buey
2010	Tigre
2011	Liebre

La función pedida recibe como parámetro un año y debe retornar el animal asociado con ese año. Verifique que su función funcione correctamente para cualquier año mayor o igual a uno, no solo para los enumerados en la tabla.

```
def yearToAnimal(year):
```

Problema 4

Dado un entero $n \geq 0$, determinar si n es un número de Fibonacci. La secuencia de Fibonacci es:

0 1 1 2 3 5 8 13 21 34 55 ...

En esta secuencia infinita el primer número es cero, el segundo es uno, y todos los demás se obtienen sumando los dos números que le preceden. Por ejemplo, el número siguiente al 55 es $89 = 34 + 55$.

La función recibe n como parámetro y debe retornar True o False para indicar si n es un número de Fibonacci o no.

```
def isFibonacci(n):
```

Problema 5

Reportar el nombre de una figura geométrica dado su número de lados, según la siguiente tabla:

Número de lados	Tipo de figura
3	triángulo
4	cuadrilátero
5	pentágono
6	hexágono
7	heptágono
8	octágono
9	eneágono
10	decágono

La función recibe n como parámetro y debe retornar el string correspondiente, en minúsculas, según la tabla anterior. En caso de que n sea mayor que 10 o menor que 3, se debe retornar None.

```
def shapeType(n):
```

Problema 6

Validar si una contraseña cumple los siguientes requisitos:

1. Tiene al menos 8 caracteres y no tiene espacios
2. Tiene al menos 3 de los siguientes tipos de caracteres:
 - a. Letras mayúsculas
 - b. Letras minúsculas

- c. Dígitos
- d. Símbolos (por ejemplo, '#', '@', '?', '\$', ...)

La función requerida recibe una contraseña como parámetro (una cadena de caracteres) y retorna *true* si esa contraseña cumple los dos requisitos mencionados. Retorna *false*, de lo contrario.

```
def isStrongPassword(pass):
```

Problema 7

Escriba una función que tome tres números como parámetros y devuelva como resultado el valor de la mediana de esos números. El valor de la mediana es el número que queda en el medio de los tres valores cuando se ordenan en orden ascendente.

```
def median(a,b,c):
```

Problema 8

El valor de π se puede aproximar mediante la siguiente serie infinita:

$$\pi \approx 3 + \frac{4}{2 \times 3 \times 4} - \frac{4}{4 \times 5 \times 6} + \frac{4}{6 \times 7 \times 8} - \frac{4}{8 \times 9 \times 10} + \frac{4}{10 \times 11 \times 12} - \dots$$

Escriba una función que reciba un entero $n \geq 1$, y retorne la aproximación a π que resulta de sumar los n primeros términos de la serie infinita anterior.

```
def approximatePI(n):
```

Problema 9

Escriba una función que genere una contraseña aleatoria. La longitud de la contraseña es el parámetro de la función, el cual debe ser un entero positivo. Cada carácter debe seleccionarse al azar de las posiciones 33 a 126 en la tabla ASCII. Devolverá la contraseña generada aleatoriamente como único resultado.

Sugerencia: probablemente encontrará útil la función **chr** al completar este ejercicio. La información detallada sobre la función **chr** está disponible en línea.

```
def randomPassword(size)
```

Problema 10

Escriba una función que determine cuántos días hay en un mes en particular. Su función toma dos parámetros: el mes como un número entero entre 1 y 12, y el año como un número entero de cuatro dígitos. Asegúrese de que su función informe el número correcto de días en febrero para los años bisiestos.

```
def daysInMonth(m,y)
```