

## EJERCICIOS RESUELTOS 5

1. Construye una calculadora rudimentaria que sume, reste, multiplique o divida.
  - a. Crear una función para escribir en pantalla un menú de opciones y devolver la opción elegida. Por ejemplo, presentar un menú indefinidamente mientras la opción no sea 1, 2, 3, 4, ó 5.
  - b. Construye un programa que invoque a la función anterior y solicite dos números por teclado simulando una calculadora rudimentaria.
2. Calcular el número combinatorio  $\binom{m}{n}$  sabiendo que m debe ser mayor o igual que n.
  - a. Usar la función factorial de la librería math.
  - b. Crear una función para calcular un factorial.
3. Crear una función para calcular  $\sum_{i=1}^n \frac{1}{i!}$ , tal que n es el argumento de entrada y vale 10 por defecto.
4. Crear una función para saber si un número es positivo. Se recibe un número y se devuelve un booleano. A continuación, construir un procedimiento que pida 3 números por teclado e invoque a la función anterior escribiendo en pantalla los positivos.
5. Crear una función para dividir dos números. Si el divisor es cero devolver un mensaje que lo indique.
6. Crear una función para calcular la media aritmética de una lista de números.
7. Crear una función para contar la cantidad de positivos, negativos y ceros de una matriz. Se recibe una lista de listas y se devuelve una tupla (positivos, negativos, ceros).
8. Crear una función para calcular el área de un cilindro. Se reciben el radio y altura del cilindro.
9. Construye dos procedimientos:
  - a. Un procedimiento `desplaza_derecha` que desplace los valores de las componentes de una lista, por ejemplo: `A[5,10,15,20,25,30,35]` una posición hacia la derecha de modo que el valor de la última componente pase a la primera, es decir, después del desplazamiento, la lista es `A[35,5,10,15,20,25,30]`.
  - b. Un procedimiento principal que reciba una lista y cuántos desplazamientos e invoque al procedimiento anterior

desplaza\_derecha tantas veces como desplazamientos quieran hacerse escribiendo la lista resultante en pantalla.

10. Construye una función en Python que encuentre todos los números reproductores de Fibonacci de dos y tres dígitos.

A saber: un número  $n$  se dice reproductor de Fibonacci si es capaz de reproducirse a sí mismo en una secuencia generada con los  $m$  dígitos del número en cuestión (sin alterar su orden) y continuando en la serie con un número que es la suma de los  $m$  términos precedentes.

Ejemplos:

- 47 es un número reproductor de Fibonacci pues la serie:  
4, 7, 11, 18, 29, 47, ... contiene el 47.
  - 13 no es un número reproductor de Fibonacci pues la serie:  
1, 3, 4, 7, 11, 18, ... no contiene el 13.
  - 197 es un número reproductor de Fibonacci pues la serie:  
1, 9, 7, 17, 33, 57, 107, 197, ... contiene el 197.
  - 985 no es un número reproductor de Fibonacci pues la serie:  
9, 8, 5, 22, 35, 62, 119, 216, 397, 732, 1345, ... no contiene el 985.
- a. Construye una función `es_repFib` para comprobar si un número de cualquier longitud es reproductor de Fibonacci. La función recibe un número y devuelve un booleano.
- b. Construye una función `repFib2_3` para explorar todos los números reproductores de Fibonacci de dos y tres dígitos. La función no recibe nada y devuelve cuantos son y cuáles.

11. Crear dos funciones en Python para mezclar y ordenar dos listas ordenadas con dos métodos diferentes.

- a. `mezclando` es una función que tiene como argumentos de entrada dos listas ordenadas con datos numéricos, y devuelve una única lista con todos sus elementos ordenados. Usar dos índices para recorrer las listas de entrada, no modificar las listas de entrada en el proceso.
- b. `ordena_mezclando` es otra función que realiza la misma tarea que la anterior, pero comparando siempre los elementos de las posiciones 0 de ambas listas de entrada, es decir, eliminando los elementos de las listas a medida que se van volcando en la lista de salida.

### **Ejemplo de presentación**

```
>>>mezclando([-1,9,9,21],[3.1,7,23,30,50])  
[-1,3.1,7,9,9,21,23,30,50]
```

```
>>>ordena_mezclando([-1,9,9,21],[3.1,7,23,30,50])  
[-1,3.1,7,9,9,21,23,30,50]
```