

Práctico 9 - Recursión

1. Implementar la función factorial de un número, de modo recursivo. Recordar que factorial de un número n se denota por $n! = 1.2.3...(n-1).(n)$ y representa el producto desde 1 hasta n . Además, se define $0! = 1$
2. Implementar una función recursiva, que tome un parámetro numérico n y devuelva la suma desde 1 hasta n .
3. Implementar una función recursiva llamada potencia que tome dos parámetros numéricos, una base b y una potencia p , y devuelva b^p
4. Implementar de modo recursivo, la función de Fibonacci. A continuación se les recuerda su definición matemática:

$$\begin{aligned}f_0 &= 0 \\f_1 &= 1 \\f_n &= f_{n-1} + f_{n-2}\end{aligned}$$

Historia: Esta sucesión fue descrita en Europa por Leonardo de Pisa, matemático italiano del siglo XIII también conocido como Fibonacci. Tiene numerosas aplicaciones en ciencias de la computación, matemática y teoría de juegos. Fibonacci, nació en 1170 y murió en 1240. Fibonacci presentó la sucesión como la solución a un problema de la cría de conejos, en su libro Liber Abaci, publicado en 1202

5. Implementar una función recursiva que tome una lista como parámetro y devuelva la suma de todos sus elementos. **Tip: Utilizar rebanadas, pedir explicación de las mismas al profe.*
 6. Implementar una función recursiva llamada *invertir* que tome una frase como parámetro y la devuelva invertida.
 7. Implementar una función recursiva que tome como parámetro una lista de nombres y una letra. Y devuelva la cantidad de nombres que empiezan con dicha letra.
 8. Implementar una función recursiva llamadas *pares* que tome una lista de números como parámetro y devuelva una lista de valores lógicos (*True / False*) donde *True* significa ese número es par en la lista de números, y *False* que no lo es. Ejemplo de lo que me quedaría de resultado para una lista inventada de números:
-
- ```
nros = [1, 4, 76, 3, 5, 8]
resultado = [False, True, True, False, False, True]
```
- 
9. Implementar una función recursiva llamada *max* que tome una lista como parámetro y devuelva el valor máximo de la lista.
  10. Implementar una función llamada *min* que tome una lista como parámetro y devuelva el valor mínimo de la lista, utilizando recursión en su resolución.
  11. Crear una función recursiva, que calcule el máximo común divisor de dos números  $a, b$ .