

Ejercicios 2.3 Recursividad

[Ejercicio 1](#)[Ejercicio 2](#)[Ejercicio 3](#)[Ejercicio 4](#)[Ejercicio 5](#)

Ejercicio 1

Realiza una función recursiva que reciba la base y el exponente y devuelva la potencia $\text{base}^{\text{exponente}}$.

Ejercicio 2

El máximo común divisor de los enteros a y b es el entero más grande que es divisor exacto de a y de b. Por ejemplo, 12 es el mcd de 36 y 60.

Escribe una función recursiva llamada mcd que devuelva el máximo común divisor de a y b según el algoritmo de Euclides:

- si $b = 0 \rightarrow \text{mcd}(a, b) = a$.
- si $b \neq 0 \rightarrow \text{mcd}(a, b) = \text{mcd}(b, a \% b)$.

Ejercicio 3

En matemáticas, la sucesión o serie de Fibonacci es la siguiente sucesión infinita de números naturales:

0, 1, 1, 2, 3, 5, 8, 13, 21, 34,

La serie comienza con los números 0 y 1 y a partir de éstos, cada término es la suma de los dos anteriores.

Realiza una función recursiva que devuelva el elemento enésimo de la serie de Fibonacci. Es decir, si recibe:

- 0: devuelve 0
- 1: devuelve 1
- 4: devuelve 3
- 7: devuelve 13

Ejercicio 4

Realiza una función recursiva que encuentre el primer valor n para el que la suma $1 + 2 + 3 + \dots + n$ exceda a un valor m. Es decir, si m vale:

- 1: devuelve 2
- 3: devuelve 3

- 7: devuelve 4
- 10: devuelve 5
- 15: devuelve 6

Ejercicio 5

Realiza un programa para resolver el juego de las **Torres de Hanoi**. El juego consiste en tres varillas verticales. En una de las varillas se apila un número indeterminado de discos. Los discos se apilan sobre una varilla en tamaño decreciente. No hay dos discos iguales, y todos ellos están apilados de mayor a menor radio en una de las varillas, quedando las otras dos varillas vacantes. El juego consiste en pasar todos los discos de la varilla ocupada a una de las otras varillas vacantes. Para realizar este objetivo, es necesario seguir estas simples reglas:

- Solo se puede mover un disco cada vez.
- Un disco de mayor tamaño no puede descansar sobre uno más pequeño que él mismo.
- Solo puedes desplazar el disco que se encuentre arriba en cada varilla.

El movimiento de n discos se puede visualizar en términos de mover sólo $n-1$ discos (y de ahí la recursión) como sigue:

1. Pasar $n-1$ discos de la varilla 1 a la 2, usando la varilla 3 como área de retención temporal.
2. Pasar el último disco (el más grande) de la varilla 1 a la 3.
3. Pasar los $n-1$ discos de la varilla 2 a la 3, empleando la varilla 1 como área de retención temporal.

El proceso termina cuando la última tarea implica pasar $n=1$ disco, esto es, el caso base. Esto se logra transfiriendo el disco sin necesidad de un área de retención temporal. Escribe una función recursiva con cuatro parámetros:

1. El número de discos por transferir.
2. La varilla en la que están colocados inicialmente esos discos.
3. La varilla a la que debe pasarse esa pila de discos.
4. La varilla que se usará como área de retención temporal.

El programa deberá imprimir las instrucciones precisas requeridas para pasar los discos de la varilla inicial a la varilla de destino. Por ejemplo, para pasar una pila de tres discos de la varilla 1 a la varilla 3, el programa deberá imprimir la siguiente serie de movimientos:

```
1 → 3
1 → 2
3 → 2
1 → 3
2 → 1
2 → 3
1 → 3
```

4/4/2019