

## Ejercicios

1. Definir la función **tercer-angulo** que reciba los valores de dos de los ángulos interiores de un triángulo y devuelva el valor del restante.
2. Definir la función **segundos** que reciba los cuatro valores (días, horas, minutos y segundos) del tiempo que dura un evento y devuelva el valor de ese tiempo expresado solamente en segundos.
3. Definir la función **sig-mul-10** que reciba un número entero y devuelva el primer múltiplo de 10 que lo supere.
4. Definir las funciones **red**, **green**, **blue** y **alpha** que reciban el valor numérico de un color de 32 bits y devuelvan, respectivamente, los valores de las componentes rojo, verde, azul y alfa (*RGBA: red, green, blue, alpha*) del mismo.
5. Definir la función **capicua?** que reciba un número entero no negativo de hasta 5 dígitos y devuelva true si el número es capicúa; si no, false.
6. Definir la función **aprox-pi** que reciba la cantidad de términos a considerar entre los paréntesis de la expresión  $4 \cdot (1 - 1/3 + 1/5 - 1/7 + \dots + 1/n)$  y devuelva la correspondiente aproximación de  $\pi$ .
7. Definir la función **invertir** que reciba un número entero no negativo y lo devuelva espejado.
8. Definir la función **nth-fibo** que reciba un número entero no negativo y devuelva el correspondiente término de la sucesión de Fibonacci.
9. Definir la función **cant-dig** que reciba un número entero y devuelva la cantidad de dígitos que este tiene.
10. Definir la función **pot?** que reciba dos números naturales y devuelva true si el primero es potencia del segundo; si no, false.
11. Definir la función **digs** que reciba un número y devuelva una lista con sus dígitos.
12. Definir la función **repartir** que, llamada sin argumentos, devuelva la cadena "Uno para vos, uno para mí". De lo contrario, se devolverá una lista, en la que habrá una cadena "Uno para X, uno para mí" por cada argumento X.
13. Definir una función para producir una lista con los elementos en las posiciones pares de dos listas dadas.
14. La transcripción es el proceso en el que la secuencia de ADN de un gen se copia (transcribe) para hacer una molécula de ARN. La cadena de ARN transcrita se forma reemplazando cada nucleótido del ADN por su complemento de ARN:  $G \rightarrow C$ ,  $C \rightarrow G$ ,  $T \rightarrow A$  y, por último,  $A \rightarrow U$ . Definir la función **adn2arn** que reciba una cadena de ADN y la devuelva transcrita en ARN.

15. Definir una función para eliminar las ocurrencias de un dato escalar en una lista (a todo nivel).
16. Definir una función para obtener el último símbolo de una lista (a todo nivel).
17. Definir una función para obtener el elemento central de una lista.
18. Definir una función para eliminar los elementos repetidos de una lista simple.
19. Definir una función para ordenar una lista de listas por longitud creciente.
20. Un ISBN-10 es válido si sus 10 dígitos  $x_1, x_2, x_3, \dots, x_{10}$  cumplen lo siguiente:  

$$(x_1 * 10 + x_2 * 9 + x_3 * 8 + x_4 * 7 + x_5 * 6 + x_6 * 5 + x_7 * 4 + x_8 * 3 + x_9 * 2 + x_{10} * 1) \bmod 11 == 0$$

Un ISBN-10 está dividido en cuatro partes: el código de país o lengua de origen (de 1 a 5 dígitos), el editor, el número del artículo y un dígito de control. Opcionalmente, estas cuatro partes pueden estar separadas mediante espacios en blanco o guiones. El dígito de control puede valer **X** que representa el valor **10**. Por ejemplo, **3-598-21507-X** es un ISBN-10 válido. Escribir la función **isbn-10?** que devuelve **true** si la cadena recibida es un ISBN-10 válido; si no, **false**.
21. Definir una función para obtener la matriz triangular superior (incluyendo la diagonal principal) de una matriz cuadrada que está representada como una lista de listas.
22. Definir una función para obtener la diagonal principal de una matriz cuadrada que está representada como una lista de listas.
23. Definir una función para transponer una lista de listas.
24. Definir una función que cuente las apariciones de cada nucleótido en una cadena de ADN.
25. Definir una función que cuente las apariciones de cada palabra en una frase.
26. Definir una función que reciba un número **n** y devuelva el enésimo número primo.
27. Definir una función que determine si una palabra tiene o no letras repetidas.
28. Definir la función **b** que aplicada a una lista de funciones **f** y a un argumento **x** obtenga la lista de resultados de aplicar cada función **f** a **x**.
29. Definir una función que devuelva el tipo de triángulo que constituyen tres números que representan longitudes.
30. Definir la función **slice** que reciba una cadena **cad** y un número **n** y devuelva una lista con todas las subcadenas contiguas de **cad** cuyo tamaño sea **n**. Por ejemplo:  

$$(\text{slice } \text{"abcde"} \ 3) \rightarrow (\text{"abc"} \ \text{"bcd"} \ \text{"cde"})$$
31. Definir una función que reciba una palabra y un conjunto de sus posibles anagramas, y devuelva un conjunto que solo contenga las palabras que realmente son sus anagramas.

32. Definir una función que reciba una cadena y devuelva el acrónimo correspondiente.
33. Definir una función que devuelva **true** si una frase es un *pangrama* (es decir, si contiene todas las letras del alfabeto); si no, **false**. Por ejemplo:  
(pangrama? "Fabio me exige, sin tapujos, que añada cerveza al whisky") → true
34. Definir la función **narcissistic?** que devuelva **true** si un número dado es igual a la suma de cada uno de sus dígitos elevado a la cantidad de dígitos; si no, **false**. Por ej:  
(narcissistic? 153) → true
35. Definir las funciones **filas-max-V** y **mas-V-o-F** que, aplicadas a una matriz de **V** y **F** (una lista de listas con los valores **V** y **F**), devuelvan, respectivamente:  
a) El/los número/s de la/s fila/s en la/s que la cantidad de **V** es máxima, por ejemplo:  
(filas-max-V '((V F V V F)(V V F V V)(F F F V F)(V V V F V))) → (2 4)  
b) **V** si en la mayoría de las filas hay más **V** que **F** o, de lo contrario, **F**, por ejemplo:  
(mas-V-o-F '((V F V V F)(F F F V F)(V V F F V))) → V
36. Definir la función **sublist** que devuelva la sublista correspondiente a una lista, una posición inicial y una longitud dadas. Por ejemplo:  
(sublist '(A B C D E F G) 3 2) → (C D)
37. Definir la función **pos-inicial** que, dadas dos listas simples, devuelva la posición inicial en la cual la primera lista se encuentra contenida en la segunda. Si no se encuentra contenida, la función devolverá cero. Por ejemplo:  
(pos-inicial '(C D) '(A B C D E F)) → 3  
(pos-inicial '(A C) '(A B C D E F)) → 0
38. Definir una función que actúe como la instrucción **distl** de FP. Por ejemplo:  
(distl 'a '(b c d)) → ((a b)(a c)(a d))
39. Definir la función **col-par-fil-imp** que devuelva la matriz resultante de tomar de una matriz las columnas pares y, de estas, las filas impares. Por ejemplo:  
(col-par-fil-imp '((1 5 7 9)(2 4 8 4)(3 6 7 8)(6 8 7 3))) → ((5 9)(6 8))
40. Definir la función **dif-sumas** que devuelva la sumatoria de los números de las filas impares menos la sumatoria de los números de las filas pares de una matriz representada mediante listas. Por ejemplo:  
(dif-sumas '((1 5 7)(2 4 8)(3 6 7))) → (1+5+7+3+6+7)-(2+4+8) → 15
41. Definir una función que devuelva la lista que resulta al intercalar los elementos de otras dos listas que recibe como parámetros. Por ejemplo:  
(intercalar '(1 2 3) '(4 5 6)) → (1 4 2 5 3 6)
42. Definir una función que calcule la profundidad de una lista. Por ejemplo:  
(profundidad '((2 3)(3 ((7))) 5)) → 4

43. Definir una función que devuelva un vector con los números primos menores que 100.
44. Definir una función que reciba un número  $n$  y devuelva un vector de vectores que tenga la siguiente forma: `[[1] [1 2] [1 2 3] [1 2 3 4] ... [1 2 3 4 ... n]]`
45. Definir una función que reciba una lista de pares ordenados, en los cuales la primera componente indica el equipo que resultó ganador de un partido y la segunda indica el perdedor (no hay empates), y que devuelva una lista con los equipos invictos.
46. Definir una función que reciba una lista de pares ordenados, en los cuales la primera componente indica el equipo que resultó ganador de un partido y la segunda indica el perdedor (no hay empates), y devuelva una lista con los equipos que solo perdieron.
47. Definir una función que reciba una lista de pares ordenados, en los cuales la primera componente indica el equipo que resultó ganador de un partido y la segunda indica el perdedor (no hay empates), y que devuelva una lista que contenga tres sublistas: en la primera aparecerán los equipos que ganaron más veces de las que perdieron; en la segunda, los equipos que perdieron más veces de las que ganaron y, en la tercera, los equipos que ganaron y perdieron la misma cantidad de veces.
48. Definir una función que devuelva **true** si una frase es un *palíndromo* (es decir, si se lee igual al derecho o al revés); si no, **false**. No se debe distinguir entre mayúsculas y minúsculas, ni considerar los acentos. Los espacios y los signos de puntuación deben ignorarse. Por ejemplo:  
(palindromo? "La ruta nos aportó otro paso natural.") → **true**
49. Definir las funciones **ieee-754** que reciba un número y devuelva una lista con los 32 bits correspondientes a su representación en notación de punto flotante según la norma IEEE-754 (*single*) y **bin2hex** que reciba una lista de bits y devuelva una cadena con su representación en hexadecimal.
50. Definir una función que reciba un número entero y devuelva una cadena con su representación en números romanos.