

Programación Técnica y Científica

Ejercicios de listas en python

1. Escribe una función `sumNumsLista(numeros)` que sume todos los números de una lista. Compara el tiempo entre usar o no **range**
2. Escribe una función `contarNumerosImpares(numeros)` que contabilice la cantidad de número impares que hay en una lista.
3. Escribe una función `numerosPares(numeros)` que devuelva los números pares que hay en una lista.
4. Escribe una función `numerosPares(numeros)` que devuelva los números pares que hay en una lista y la suma de esos elementos.
5. Escribe una función `maximo(numeros)` que devuelva el elemento más grande de una lista.
6. Dada una matriz representada como una lista, escribe una función `dims(lista)` que devuelva sus dimensiones.
7. Escribe una función `combinar(la, lb)` que dadas dos listas devuelva una lista conteniendo los elementos de ambas listas ordenados de forma ascendente.
8. La traspuesta de una matriz se obtiene intercambiado filas y columna. Escribe una función que devuelva la traspuesta de una matriz.
9. Escribe una función `factores_primos(num)` que devuelva una lista con la descomposición en factores primos de num
10. Escribe una función `sumaAcumulada(numeros)` a la que se le pase una lista de números y devuelva una lista en la que el elemento i-ésimo se obtiene como la suma de los elementos de la lista entre las posiciones 0 e i. Por ejemplo, para [1, 2, 3] sería [1, 3, 6]
11. Escribe una función `combinarListas(l1, l2)` que devuelva una lista que esté formada por todos los elementos de l1 y a continuación todos los de l2. Por ejemplo `combinaLista([1, 2, 8], [5, 10])` devolvería [1, 2, 8, 5, 10]
12. Escribe una función `eliminar(l1, l2)` que dadas dos listas devuelva una lista en la que estén todos los elementos de l1 que no están en l2.
13. Escribe una función `contarLetras(palabra)` que tome una palabra como argumento y devuelva una lista de pares en la que aparece cada letra junto con el número de veces que aparece esa letra en la palabra. Por ejemplo, `contarLetras('patata')` devuelve [('a', 3), ('p', 1), ('t', 2)]
14. Escribe una función `longs(cadenas)` a la que le pasa una lista de cadenas y devuelve una lista con las longitudes de cada una de ellas.
15. Escribe una función `cadena_mas_larga(cadenas)` a la que se pasa una lista de palabras y que devuelva la palabra más larga.
16. Escribe una función `cadena_mas_larga(cadenas, n)` a la que se pasa una lista de palabras y un número y que devuelva las palabras con longitudes mayores que n.
17. Escribe una función `obtenerPotencias(numero)` que devuelva una lista del siguiente tipo [numero², (numero-1)²,..., 0, 1², 2²,... numero²]

18. Escribe una función `sumaPrimerDigito(numeros)` que devuelva la suma de los primeros dígitos de todos los números de la lista que se pasa como argumento.
19. Escribe una función `mayoresMedia(numeros)` que devuelva una lista con los números que son mayores que la media. Usa comprensión de listas.
20. Escribe una función que realice la multiplicación de matrices. Dadas dos matrices (representadas como listas anidadas) de tamaño **mxn** y **nxp** devuelva una de tamaño **mxp**
$$A_{m,n} = [[x_{11}, x_{12}, \dots, x_{1n}], \dots, [x_{m1}, \dots, x_{mn}]]$$
21. Un vector disperso es aquel que tiene muchos elementos nulos. Para ese tipo de vectores, la representación más adecuada es guardar únicamente los elementos no nulos. Escribe una función `dispersa(v)` a la que se le pase una lista representando un vector disperso y que devuelva una lista de pares (pos, elem) con cada una de las posiciones en las que hay un elemento no nulo y el elemento.
22. Resuelve el mismo problema anterior pero para una matriz dispersa y devolviendo (i, j, elem)
23. Escribe una función `vector(v_disperso)` que dada una representación compacta del vector cree un vector con los elementos nulos.
24. Resuelve el mismo problema del ejercicio anterior pero para una matriz.