PRÁCTICAS PROGRAMACIÓN TÉCNICA Y CIENTÍFICA

SESIÓN 4. Listas

En esta sesión usaremos **el tipo lista**, se deben definir funciones, se pueden usar sentencias de E/S, condicionales, bucles así como el tipo string y el resto de tipos de datos básicos de python. Además, hay que tener en cuenta que esta clase tiene ya métodos para realizar algunas de estas tareas, pero la idea es realizar los ejercicios **primero evitando estos métodos propios de list**, y luego, en una **segunda versión** del ejercicio volver a realizarlo con dichos métodos. Es decir, en la primera versión, hay que recorrer la lista, en cada ejercicio, de modo que **no se pueden usar** los métodos count(), index(), remove(), min(), max(), sum(), sort(), reverse(), ni similares. En la segunda versión de la solución sí los podéis usar. En cada ejercicio hay que hacer las dos versiones, siempre que exista un método que lo solucione.

1. Crea una lista con los valores enteros de 1 a N e implementa una función que reciba dicha lista y nos devuelva la suma de dichos valores. Solicitar N por teclado y mostrar el resultado por pantalla.

2. Crea una lista con los valores enteros de 1 a N e implementa una función que reciba dicha lista y nos devuelva una lista con los valores impares y el número de dichos valores. Solicitar N por teclado y mostrar el resultado por pantalla.

3. Crea una lista con los valores enteros de 1 a N e implementa una función que reciba dicha lista y nos devuelva el máximo y el mínimo de dichos valores, así como sus respectivas posiciones. Solicitar N por teclado y mostrar el resultado por pantalla.

4. Usando el módulo “random” genera dos listas de N y M números enteros aleatorios entre 1 y 10 e implementa una función que devuelva una tercera lista que contenga los números de las dos primeras listas en orden ascendente sin contener valores repetidos. Solicitar N y M por teclado y mostrar el resultado por pantalla.

5. Partiendo de una lista que contiene a su vez N listas de M enteros, si la consideramos como una matriz de dimensión NxM, implementar una función que nos devuelva la matriz traspuesta MxN (intercambiando filas y columnas) que contedrá M listas de N enteros. Solicitar N y M por teclado y mostrar el resultado por pantalla.

6. Solicitar un número entero N por teclado e implentar una función que devuelva una lista con la descomposición en factores primos de N. Mostrar el resultado por pantalla.

7. Usando el módulo “random” genera dos listas de N y M números enteros aleatorios entre 1 y 10 e implementa una función que devuelva una tercera lista que represente la intersección de las dos primeras. Los valores deben estar ordenados en orden ascendente. Solicitar N y M por teclado y mostrar el resultado por pantalla.

8. Realizar un programa que muestre un menú de opciones al usuario con las siguientes operaciones sobre una lista que inicialmente está vacía: 1) Insertar un entero positivo (debe insertarlo en orden ascendente), 2) Eliminar un valor de la lista dado el entero positivo, 3) Eliminar un valor dada su posición, 4) Salir. Después de cada operación se debe siempre mostrar al usuario el estado de la lista y se deben controlar las posibles situaciones de error informando al usuario de dicho error y volviendo a solicitar la entrada correspondiente.

9. Usando el módulo “random” genera dos listas de N y M números enteros aleatorios entre 1 y 10 e implementa una función que devuelva una tercera lista con los valores de la primera que NO estén en la segunda. Los valores deben estar ordenados en orden ascendente. Solicitar N y M por teclado y mostrar el resultado por pantalla.

10. Leer una frase de teclado e implementar una función que devuelva una lista de pares en la que debe aparecer cada letra junto a su frecuencia de aparición. Los espacios no se deben tener en cuenta. Dicha lista debe estar ordenada atendiendo al orden ascendente de las letras. Ejemplo: ante la entrada “programa” debe dar como salida [('a', 2), ('g', 1), ('m',1), ('o', 1), ('p',1), ('r',2)].