

### Ejercicios - Listas y Tuplas

1. Escribir una función que reciba por parámetro una tupla de números enteros. La función deberá devolver 1, si la tupla se encuentra ordenada en forma creciente; -1 si la tupla está ordenada en forma decreciente; ó 0 si la tupla está desordenada.
2. Escribir un programa que genere una tupla de 100 valores aleatorios. Cada uno de los valores, debe ser entero, y estar entre 0 y 10000, inclusive; y sólo debe aparecer una vez. Luego, se deben listar por pantalla los números que sean primos e informar cuántos son, y la suma acumulada de los mismos.
3. Escribir una función que reciba un entero y retorne una lista con los valores de la [sucesión de Fibonacci](#) desde la posición 0 y hasta la posición correspondiente al parámetro recibido.  
Recordemos que la sucesión comienza con los número 0 y 1; y a partir de estos, cada término es la suma de los dos anteriores.
4. Escribir un programa que solicite al usuario el ingreso de un número natural N, y luego informe, primero los números primos que hay entre 0 y el número ingresado, y luego los números que no son primos. Para la solución, debe implementar el método de la [Criba de Eratóstenes](#), que permite hallar los números primos menores que un número natural dado.  
En base a una lista con todos los números naturales comprendidos entre 2 y N, se descarta primero el 2 y todos sus múltiplos. Luego se repite el proceso para el primer número que aparece en la lista y todos sus múltiplos. Así sucesivamente hasta que el cuadrado del primer número que aparece como primero en la lista resultante, sea mayor a N.
5. Escribir una función que indique si los 5 dados que han sido arrojados por un jugador, forman escalera. La función recibirá por parámetro, una tupla, con los 5 valores obtenidos al arrojar los dados. Deberá devolver True, si forman escalera, de lo contrario, deberá devolver, False.  
Para que se de **escalera**, hay 3 alternativas: (1,2,3,4,5); (2,3,4,5,6) ó (3,4,5,6,1), claro que el orden en el que aparecen los valores, no importa.
6. De igual modo que el ejercicio anterior, escribir una función para cada uno de los posibles casos:  
**Generala**: 5 dados de igual valor  
**Póker**: 4 dados iguales y 1 distinto  
**Full**: 3 dados iguales y otros 2 iguales
7. Ahora vamos a escribir un programa que simule una situación de juego, utilizando las funciones escritas en los puntos 3 y 4; y sumando nuevas funciones para lo que se solicite en este punto específicamente.  
Escribir un programa que genere 1000 tiradas aleatorias, de los 5 dados.  
Por cada tirada, se debe clasificar si corresponde a alguna de las categorías para las que hemos escrito las funciones, ó a ninguna de ellas.  
El programa deberá emitir un informe que indique la cantidad de tiradas coincidentes con cada categoría (incluida que no coincide con ninguna categoría), y el % que representa cada una.

8. Escribir un programa que genere una lista anidada, con 50 sublistas de pares de valores aleatorios. Los valores deben estar entre -1000 y 1000, inclusive.  
Luego, agregar a cada sublista de pares, un tercer elemento, que sea el MCD de los otros dos.  
Por último, suprimir de la lista, las ternas cuyo MCD sea igual a 1.  
Listar las ternas resultantes.
9. Escribir una función que reciba una tupla de N números enteros.  
La función deberá devolver una lista formada por pares de elementos, donde el primer elemento, será uno de los valores que venía en la tupla, y el segundo elemento, la cantidad de veces que dicho valor aparecía en la tupla. La lista debe estar ordenada por el primer elemento del par.  
Los valores sólo deben aparecer una única vez en la lista.  
Pruebe su función, con tuplas generadas aleatoriamente:
  - a. Con 100 elementos, con valores entre 0 y 10.
  - b. Con 1000 elementos, con valores entre -10 y 10.
  - c. Con 5000 elementos, con valores pares entre 2 y 20.
10. Si para el ejercicio anterior, usaste el método count; y eliminaste a través del método pop ó con la función del(), los valores que aparecían más de una vez; reescribí la función pero sin utilizar ninguna de estas alternativas.
11. Necesitamos elegir al azar una persona, dentro de un grupo de personas. Por ello escribiremos un programa que nos permita realizar esta operación.  
El programa deberá solicitar el ingreso de los nombres y apellidos de cada uno de los integrantes de este grupo. El ingreso de nombres finalizará cuando el usuario en lugar de ingresar un nombre y apellido, simplemente, de enter.  
A medida que se ingresan los nombres y apellidos, se debe controlar que el que se está ingresando, no haya sido ingresado.  
Una vez finalizada la carga, el programa deberá informar cual es el elegido.
12. Escribir un programa que solicite el ingreso de un texto formado sólo por sustantivos en singular, separados uno del otro, por un blanco.  
El usuario ingresará por ejemplo: "casa canción río bebé pez paraguas tórax"  
El programa deberá mostrar cada una de las palabras ingresadas, seguida por su palabra en plural, un par por línea (singular - plural), en orden alfabético.  
Validar que las palabras ingresadas por el usuario, sólo contengan letras; caso contrario, enviar un mensaje acorde y volver a solicitar el ingreso.  
Las reglas a cumplir para pasar un sustantivo en singular a su plural son:
  - a. Agregar una "s" al final, si la palabra termina en vocal sin acento.
  - b. Agregar una "s" al final, si la palabra termina con una é (acentuada).
  - c. Si la palabra termina en "z", la reemplazamos por "ces".
  - d. Agregamos "es" al final, si la palabra termina en una consonante (a excepción de la "s", la "z", y la "x"), ó si la palabra termina con las vocales acentuadas: á, í, ó, ú.
  - e. Si el sustantivo termina en "s" ó "x", consideramos que el plural es igual al singular, por lo tanto la función deberá devolver lo mismo que recibió.