Ejercicios Python

1. Haz una función que calcule y devuelva el número de vocales en la cadena dada. Consideraremos a, e, i, o, u como vocales. La cadena de entrada sólo consta de letras minúsculas y/o espacios.

2. Los cajeros automáticos permiten códigos PIN de 4 o 6 dígitos y los códigos PIN no pueden contener más que exactamente 4 dígitos o exactamente 6 dígitos. Si a la función se le pasa una cadena de PIN válida, devuelve true, de lo contrario devuelve false.

3. Haz una función que como parámetro reciba un array de números y obtenga el número que menos repeticiones haya tenido. En caso de empate devuelve el número más pequeño.

4. Dada un array de enteros, encuentra todo los números que aparecen un número impar de veces.

5. Implementar la función que toma como argumento una secuencia de enteros o string y devuelve una lista de elementos sin ningún elemento repetido y preservando el orden original de los elementos.

6. Escribe una función que tome un parámetro positivo num y devuelva su persistencia multiplicativa, que es el número de veces que debes multiplicar los dígitos de num hasta llegar a un solo dígito.

Por ejemplo (Entrada --> Salida):
39 --> 3 (porque 3*9 = 27, 2*7 = 14, 1*4 = 4 y el 4 sólo tiene un dígito)
999 --> 4 (porque 9*9*9 = 729, 7*2*9 = 126, 1*2*6 = 12, y finalmente 1*2 = 2)
4 --> 0 (porque el 4 ya es un número de un dígito)

7. Escribe una función que tenga como parámetro un array de números enteros. Tu trabajo es tomar esa array y encontrar un índice N en el que la suma de los enteros a la izquierda de N sea igual a la suma de los enteros a la derecha de N. Si no hay ningún índice que haga que esto ocurra, devuelve -1. Si se le da un array con múltiples respuestas, devuelve el menor índice correcto.

Digamos que te dan el array {1,2,3,4,3,2,1}:
Tu función devolverá el índice 3, porque en la 3ª posición del array, la suma del lado izquierdo del índice ({1,2,3}) y la suma del lado derecho del índice ({3,2,1}) son ambas iguales a 6.

Veamos otra.
Te dan el array {1,100,50,-51,1,1}:
Su función devolverá el índice 1, porque en la primera posición de la matriz, la suma del lado izquierdo del índice ({1}) y la suma del lado derecho del índice ({50,-51,1,1}) son ambas iguales a 1.

La última:
Se le da la matriz {20,10,-80,10,10,15,35}
En el índice 0 el lado izquierdo es {}
El lado derecho es {10,-80,10,10,15,35}
Ambos son iguales a 0 cuando se suman. (Las matrices vacías son iguales a 0 en este problema)
El índice 0 es el lugar donde el lado izquierdo y el lado derecho son iguales.

Entrada:
Un array de enteros de longitud 0 < arr < 1000. Los números del array pueden ser cualquier entero positivo o negativo.

Salida:
El índice más bajo N en el que el lado a la izquierda de N es igual al lado a la derecha de N. Si no se encuentra un índice que se ajuste a estas reglas, entonces se devolverá -1.

8. Implementa una función de diferencia, que devuelva un array que tenga todos los valores de la lista pasada como primer parámetro que no están presentes en la lista b manteniendo su orden. Si un valor está presente en b, todas sus apariciones deben ser eliminadas de la otra:

arrayDiff([1,2],[1]) == [2]
arrayDiff([1,2,2,2,3],[2]) == [1,3]

La representación binaria de 1234 es 10011010010, por lo que la función debería devolver 5 en este caso

9. Haz una función que pueda tomar cualquier número entero no negativo como argumento y devolverlo con sus dígitos en orden descendente. Esencialmente, reordenar los dígitos para crear el mayor número posible.

Entrada: 42145 Salida: 54421

Entrada: 145263 Salida: 654321

Entrada: 123456789 Salida: 987654321

10. Escriba una función que tome un número decimal como entrada, y devuelva el número de bits que son iguales a uno en la representación binaria de ese número. Comprueba que la entrada no sea negativa.

11. El teorema de los cuatro cuadrados de Lagrange, también conocido como conjetura de Bachet, afirma que todo número natural puede representarse como la suma de cuatro cuadrados enteros.

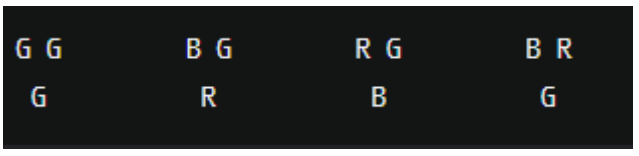
$$
\begin{aligned}
p &= a_0^2 + a_1^2 + a_2^2 + a_3^2 \\
3 &= 1^2 + 1^2 + 1^2 + 0^2 \\
31 &= 5^2 + 2^2 + 1^2 + 1^2 \\
310 &= 17^2 + 4^2 + 2^2 + 1^2 \\
&= 16^2 + 7^2 + 2^2 + 1^2 \\
&= 15^2 + 9^2 + 2^2 + 0^2 \\
&= 12^2 + 11^2 + 6^2 + 3^2.
\end{aligned}
$$

Haz una función que devuelva un array con los cuatro números naturales que cumplan el teorema dado un número natural pasado como argumento.

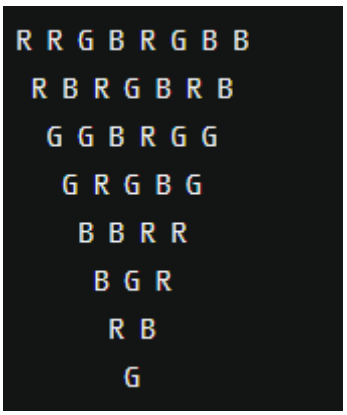
12. **Ejercicio colorear triángulo**

Un triángulo de color se crea a partir de una fila de colores, cada uno de los cuales es rojo, verde o azul. Las filas sucesivas, cada una con un color menos que la anterior, se generan considerando los dos colores que se tocan en la fila anterior.

Si estos colores son idénticos, se utiliza el mismo color en la nueva fila. Si son diferentes, se utiliza el color que falta en la nueva fila. Así se continúa hasta que se genera la última fila, con un solo color.



Un ejemplo más extenso



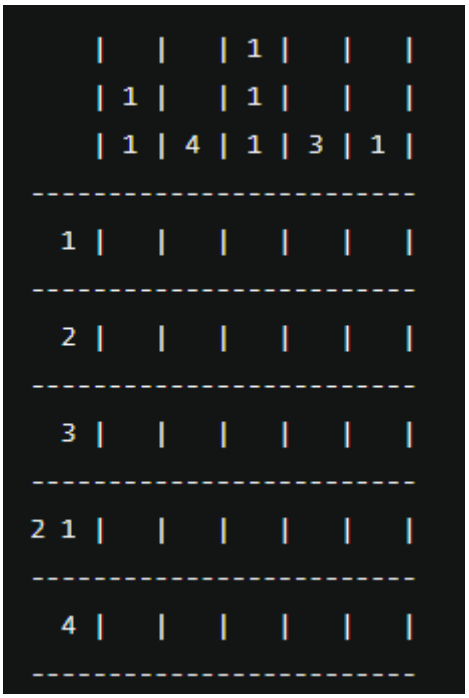
Se le dará la primera fila del triángulo como parámetro y la función debe devolver el color final que aparecería en la fila inferior como una cadena. En el caso del ejemplo anterior, se le dará "RRGBRGBB", y deberá devolver "G".

13. **Ejercicio resolución nonogramas**

Los nomogramas son rompecabezas de lógica de imágenes en los que las celdas de una cuadrícula deben ser coloreadas o dejadas en blanco según los números en el lado de la cuadrícula para revelar una imagen oculta tipo pixel art.

Se te darán las pistas y deberás devolver el puzzle resuelto. Todos los rompecabezas se podrán resolver, por lo que no tendrás que preocuparte por ello. Habrá exactamente una solución para cada puzzle.

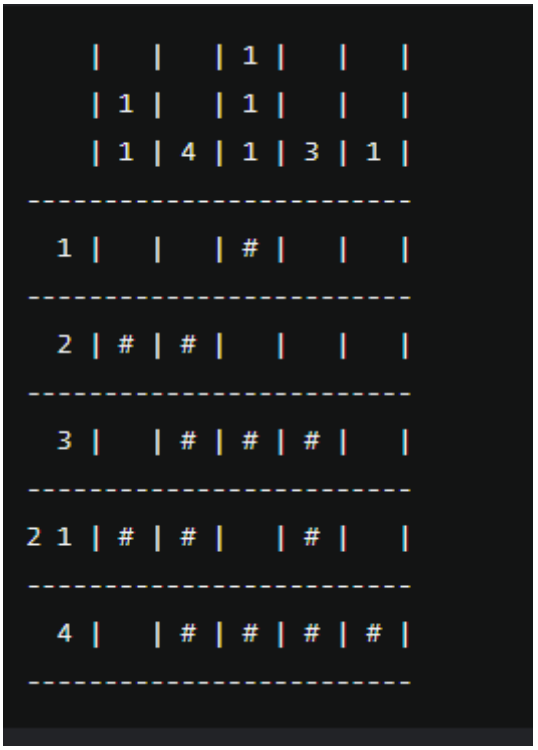
Las pistas serán una tupla de las pistas de la columna, luego las pistas de la fila, que contendrán las pistas individuales.



Las pistas están en la parte superior y en la izquierda del puzzle, así que en este caso:

Las pistas de la columna son: ((1, 1,0), (4,0,0), (1, 1, 1), (3,0,0), (1,0,0)), y las pistas de la fila son: ((1,0), (2,0), (3,0), (1, 2), (4,0)).

Las pistas de las columnas se dan de izquierda a derecha. Si hay más de una pista para la misma columna, se da primero la pista superior. Las pistas de las filas se dan de arriba a abajo. Si hay más de una pista para la misma fila, se da primero la pista más a la izquierda.



Esa sería la solución del anagrama anterior y vuestra función tendría que devolver

```
((0, 0, 1, 0, 0),
 (1, 1, 0, 0, 0),
 (0, 1, 1, 1, 0),
 (1, 1, 0, 1, 0),
 (0, 1, 1, 1, 1))
```

Rúbrica

Si un ejercicio da un error de ejecución o no es sintácticamente correcto entonces se le pondrá un 0 al ejercicio. Se valorará el uso de código que siga SOLID/KISS/DRY.

1 - 10	5
11-13	4
SOLID/KISS/DRY	1
TOTAL	10

Por cada ejercicio se dará 0,5 en el caso de las preguntas una a la diez. El ejercicio once se valorará con 1 punto y el ejercicio doce 2 puntos y trece 1.

Entregable

El entregable será la URL de vuestro repositorio de GitHub cuyo nombre será Nombre_Apellido_DAM2 donde tendréis todas las prácticas. Por cada ejercicio tendréis que hacer una clase nueva siguiendo el formato realizado por el profesor en los ejercicios de clase.

Por cada tema os iré pidiendo prácticas que tendréis que entregar utilizando GitHub.

Para esta práctica vuestro repositorio tendrá una carpeta llamada Tema_2 donde estará contenida la solución de los ejercicios. Obligatorio utilizar el .gitignore usado por el profesor.