



HOJA DE EJERCICIOS 005

Para los siguientes ejercicios, deberá desarrollarlos contemplando necesariamente lo siguiente:

- Pseudocódigo
- Diagrama de Flujo
- Código en Python

(*) Todos los ejercicios deberán ser desarrollados con los algoritmos de ordenamiento y búsqueda.

- Elaborar el algoritmo que calcule el M.C.M. y el M.C.D. de tres números ingresados por el usuario. El programa debe validar que los números sean enteros positivos. El programa se repite dos veces.
- Definir una función **generaGraficoEstadístico(listParam)** que tome una lista de números enteros e imprima un histograma en la pantalla. Considerar que la lista de parámetros puede tener N elementos.

Ej. `generaGraficoEstadístico([4, 9, 7])`

Debería imprimir:

```
****
*****
*****
```

- Elabore el algoritmo que permita obtener todas las parejas de números amigos menores o iguales que un número N ingresado por el usuario. Validar que el dato ingresado sea un número. Dos números son amigos si cada uno de ellos es igual a la suma de los divisores del otro.

Ej. 224 y 284 son amigos ya que:

- **284** : $1 + 2 + 4 + 71 + 142 = 220$
- **220** : $1 + 2 + 4 + 5 + 10 + 11 + 20 + 22 + 44 + 55 + 110 = 284$

- Elabore el algoritmo de un programa que permita crear una matriz de un número de **F** de filas y **C** de columnas indicadas por el usuario.

A dicha matriz se asignará ya sea de manera manual (por el usuario) o aleatoria (por el sistema) un símbolo que representará a un árbol en cada posición de la matriz (plano creado).

Finalmente el programa debe indicar qué porcentaje del terreno está copado de árboles de acuerdo al número de posiciones de la matriz y el número de árboles.

Ej. $F = 4, C = 3$

		*
*	*	*
	*	
*		

Considerando que el número de posiciones de la matriz es de doce, entonces el porcentaje de terreno copado por los árboles es del 50%.

- Un analista financiero lleva un registro del precio del dólar día a día, y desea saber cuál fue la mayor de las alzas en el precio diario a lo largo de ese período.



Escriba un programa que de una lista de diccionarios, que puede ser ingresado por el usuario para un número n de días, que tenga el precio del dólar para cada uno de los n días.

El programa debe entregar como salida cuál fue la mayor de las alzas de un día para el otro, indicando los días en lo que esto ocurrió.

Si en ningún día el precio subió, la salida debe decir: No hubo alzas.

```
dicPrecioDolar = [{'dia':1, 'precio': 496.96}, {'dia':2, 'precio': 499.03}, {'dia':3, 'precio': 496.03}, {'dia':4, 'precio': 493.27}, {'dia':5, 'precio': 488.82}, {'dia':6, 'precio': 492.16}, {'dia':7, 'precio': 490.32}, {'dia':8, 'precio': 490.67}, {'dia':9, 'precio': 490.89}, {'dia':10, 'precio': 494.10}]
```

Ej. Para la lista, las salidas deben ser: La mayor alza fue de 3.34 soles entre los días 5 y 6.

6. Dada la lista = [74, 53, 32, 39, 92, 8, 6, 97, 49, 75, 85, 66, 27, 25, 1, 23, 4, 17, 81, 8, 5, 24], elaborar el algoritmo y/o código python que me permita saber la ubicación (índices) exactos de los elementos que se repiten.

Ej. Dada la lista las posiciones que se repiten son: 5 y 19

7. Se dice que una matriz tiene un punto de silla si alguna posición de la matriz es el menor valor de su fila, y a la vez el mayor de su columna. Escribir un programa que tenga como entrada una matriz de números enteros, y calcular la posición de un punto de silla (si es que existe).

Ej.

- En la matriz A, el punto de silla está en la ubicación 0 , 1
- En la matriz B, el punto de silla está en la ubicación 1 , 1

$$A = \begin{bmatrix} 20 & 8 & 12 \\ 32 & 4 & 21 \\ 6 & 2 & 1 \end{bmatrix} \quad B = \begin{bmatrix} 21 & 14 & 32 & 40 \\ 53 & 15 & 34 & 45 \\ 34 & 10 & 58 & 93 \end{bmatrix}$$

8. Entre todos los enteros mayores a 1 hay solamente cuatro que pueden ser representados por la suma de los cubos de sus dígitos.

Uno de esos números es 153 pues:

$$1^3 + 5^3 + 3^3 = 1 + 125 + 27 = 153$$

Desarrolle un programa para poder determinar los otros tres números.

(*) Tenga en cuenta que los números se encuentran entre 150 y 410.

9. Escribir una función convertStrToDict(cadena) que reciba una cadena y devuelva un diccionario con la cantidad de apariciones de cada palabra en la cadena.

Ej.

ENTRADA	SALIDA
Ingrese la Cadena: "Compré pocas copas, pocas copas compré, como compré pocas copas, pocas copas pagaré."	Diccionario = {'compré': 3, 'pocas': 4, 'copas': 4, 'como': 1, 'pagaré':1}



10. Realizar el algoritmo que valide el nombre, sexo y edad de una persona. El nombre sólo debe contener caracteres del alfabeto. No se permite otro tipo de carácter o símbolos especiales. Para el sexo y la edad maneje usted un criterio lógico.
11. Realizar el algoritmo de un programa que pida que se ingrese una oración, una palabra a buscar y una palabra sustituta. El algoritmo debe reemplazar la palabra buscada por la palabra sustituta y mostrar la oración actualizada.
12. Realizar el algoritmo de un programa que pida que se ingrese una oración y una palabra a ser eliminada. El programa debe eliminar la palabra a eliminarse y mostrar la oración actualizada.
13. Escribir el algoritmo que permita crear dos listas de palabras y que a continuación elimine de la primera lista los elementos de la segunda. El programa se repite mientras el usuario lo desee.
14. Diseñe el algoritmo para generar el siguiente tablero en una matriz:

		PB
	PN	
AN		

- PN: Peón negro ,
- AN: Alfil negro,
- PB: Peón blanco.

Posteriormente el programa deberá permitir al usuario realizar un movimiento ya sea del caballo o del alfil negro a una posición válida.

Ej. de la salida:

- Qué desea mover: Peón
- A dónde:
 - o Fila:
 - o Columna:
- Movimiento realizado con éxito (si la posición a donde se quiere mover la ficha es válido)
- Movimiento inválido (si la posición a donde se quiere mover la ficha es inválido)

Finalmente el programa deberá imprimir el tablero actualizado con las fichas en sus posiciones finales.

15. Escriba un algoritmo/código python que despliegue todos los subconjuntos no vacíos del conjunto de los números del 1 a n. Por ejemplo, si n = 3, debe obtenerse: **(use recursividad)**.



1
2
3
1 2
1 3
2 3
1 2 3

16. El número de Euler, $e \approx 2,71828$, puede ser representado como la siguiente suma infinita:

$$e = \frac{1}{0!} + \frac{1}{1!} + \frac{1}{2!} + \frac{1}{3!} + \frac{1}{4!} - \dots$$

Desarrolle un función que retorne un valor aproximado de e, calculando esta suma hasta que la diferencia entre dos sumandos consecutivos sea menor que 0,0001.

(*) Recuerde que el factorial n! es el producto de los números de 1 a n.

17. Escribir una función que reciba una lista ordenada y un elemento, si el elemento se encuentra en la lista, debe encontrar su posición, mediante búsqueda binaria y devolverlo. Si no se encuentra, debe agregarlo a la lista en la posición correcta y devolver esa nueva posición.

IMPORTANTE: No utilizar lista.sort()

18. Realice dos implementaciones del algoritmo ordenación de Selección Directa aplicados sobre una lista con 100 valores aleatorios entre 1 y 1000 de manera que:
- En la primera implementación se realice la selección del elemento mayor en cada iteración
 - En la segunda implementación se realice la selección del elemento menor en cada iteración.

```
import random
random.randrange(1,100)
```

19. Programar un algoritmo **recursivo** que permita multiplicar los elementos de un vector (entendiendo que un vector es una lista unidimensional).
20. Programar un algoritmo **recursivo** que permita sumar los dígitos de un número, el mismo que deberá ser tratado en todo momento como entero.
Ej. Entrada: 123 -> Resultado: 6.
21. Realizar un programa que dada la lista = [74, 53, 32, 39, 92, 6, 97, 49, 75, 85, 66, 27, 25, 1, 23, 4, 17, 81, 5, 24], el usuario debe seleccionar un número y el sistema deberá adivinarlo, pudiendo realizar un máximo de 04 preguntas. Para ello el programa debe ir diciendo números de la lista y el humano responderá con los símbolos '<', '>' o '=', según el número pensado sea menor, mayor o igual que el dicho por el ordenador. Cuando lo adivine deberá poner un mensaje especificando cuantas preguntas ha necesitado y finalizar.



22. Se dispone de un grupo de personas que guardan como información su edad y nombre, esta información se encuentra en una lista de diccionarios con la información de cada persona. Se requiere de un programa que ordene la colección de datos por edad ascendentemente, en caso se tenga dos personas con la misma edad, se debe tomar el orden alfabético del nombre (ascendente).

LISTA DE DICCIONARIOS	SALIDA	
	Edad	Nombre
[{'nombre': 'José', 'edad': 8}, {'nombre': 'Pedro', 'edad': 12}, {'nombre': 'Alberto', 'edad': 10}, {'nombre': 'Luis', 'edad': 12}]	8	José
	10	Alberto
	12	Luis
	12	Pedro

23. Desarrollar un algoritmo que ingresado por teclado el lado **N** de la matriz cuadrada, permita generar y visualizar una matriz asimétrica.

Matriz antisimétrica es una matriz cuadrada en la que los elementos a ambos lados de la diagonal principal son opuestos (iguales pero con distinto signo) $a_{ij} = -a_{ji}$ y los elementos de la diagonal principal deben ser cero.

Ej. N=3. Los valores cargados en la matriz serán generados aleatoriamente

0	-3	-4
3	0	5
4	-5	0

```
import random
random.randrange(1,100)
```

24. Elaborar un algoritmo que ingresados por teclado el lado de una matriz cuadrada y el tipo de matriz triangular (0 = Matriz Triangular Superior, 1 = Matriz Triangular Inferior) muestre por pantalla la matriz solicitada cargándolos según corresponda con valores aleatorios.

Matriz Triangular Inferior

2	0	0
4	2	0
9	5	6

Matriz Triangular Superior

6	9	3
0	5	2
0	0	1