



DESDE LA 1:00H
MARTES 31
DE DICIEMBRE

FIN DE AÑO 2024

THE END
FABRIK
- MADRID -

Valentina Otero Fernández

RELACIÓN DE EJERCICIOS PYTHON: Nº 4

'''

1. Escriba un programa que permita crear una lista de palabras. Para ello, el programa tiene que pedir un número y luego solicitar ese número de palabras para crear la lista. Por último, el programa tiene que escribir la lista.

```
n = int(input("¿Cuántas palabras desea introducir en la lista? "))
l = []
for i in range(n):
    p = input("palabra "+str(i+1)+": ")
    l.append(p)
print(l)
```

#####
#####

2. Escriba un programa que permita crear una lista de palabras y que, a continuación, pida una palabra y diga cuántas veces aparece esa palabra en la lista.

```
n = int(input("¿Cuántas palabras desea introducir en la lista? "))
l = []
for i in range(n):
    p = input("palabra "+str(i+1)+": ")
    l.append(p)
print(l)
p = input("¿Qué palabra desea contar? ")
n = l.count(p)
print("La palabra",p,"aparece",n,"veces en ",l)
```

#####
#####

3. Escriba un programa que permita crear una lista de palabras y que, a continuación, pida dos palabras y sustituya la primera por la segunda en la lista.

```
n = int(input("¿Cuántas palabras desea introducir en la lista? "))
l = []
for i in range(n):
    p = input("palabra "+str(i+1)+": ")
    l.append(p)
print(l)
p1 = input("Introduzca la palabra a sustituir: ")
p2 = input("Introduzca la palabra sustituta: ")
l[1.index(p1)] = p2 # si la palabra no está en la lista dará error en tiempo de ejecución
print("Ahora la lista queda así:")
print(l)
```

#####
#####

4. Escriba un programa que permita crear una lista de palabras y que, a continuación, pida una palabra y elimine esa palabra de la lista.

```
n = int(input("¿Cuántas palabras desea introducir en la lista? "))
l = []
for i in range(n):
    p = input("palabra "+str(i+1)+": ")
    l.append(p)
print(l)
p = input("Introduzca la palabra a eliminar: ")
l.remove(p) # si la palabra no está en la lista dará error en tiempo de ejecución
print("Ahora la lista queda así:")
print(l)
```

#####
#####

REGRESIÓN
CLÁSICO

ENTRADAS
ANTICIPADAS
ESCANEANDO
ESTE QR.



5. Escriba un programa que permita crear dos listas de palabras y que, a continuación, elimine de la primera lista los nombres de la segunda lista.

```
n = int(input("¿Cuántas palabras desea introducir en la primera lista? "))
l1 = []
for i in range(n):
    p = input("palabra "+str(i+1)+": ")
    l1.append(p)
n = int(input("¿Cuántas palabras desea introducir en la segunda lista? "))
l2 = []
for i in range(n):
    p = input("palabra "+str(i+1)+": ")
    l2.append(p)
print("Primera lista:",l1)
print("Segunda lista:",l2)
for p in l2:
    if l1.count(p) > 0:      # sólo si la palabra tomada de la segunda lista está en la primera
        l1.remove(p)          # es eliminada
print("Primera lista filtrada:",l1)

#####
#####
```

6. Escriba un programa que permita crear una lista de palabras y que, a continuación, cree una segunda lista igual a la primera, pero al revés (no se trata de escribir la lista al revés, sino de crear una lista distinta).

```
n = int(input("¿Cuántas palabras desea introducir en la lista? "))
l = []
for i in range(n):
    p = input("palabra "+str(i+1)+": ")
    l.append(p)
print(l)
l2 = l
l2.reverse()
print(l2)

#####
#####
```

7. Escriba un programa que permita crear una lista de palabras y que, a continuación, elimine los elementos repetidos (dejando únicamente el primero de los elementos repetidos).

```
n = int(input("¿Cuántas palabras desea introducir en la lista? "))
l = []
for i in range(n):
    p = input("palabra "+str(i+1)+": ")
    l.append(p)
print(l)
for p in l:
    while l.count(p) > 1:    # mientras la palabra esté repetida
        l.remove(p)
print("lista sin repeticiones:",l)

#####
#####
```

8. Escriba un programa que permita crear dos listas de palabras y que, a continuación, escriba las siguientes listas (en las que no debe haber repeticiones):

- Lista de palabras que aparecen en las dos listas.
- Lista de palabras que aparecen en la primera lista, pero no en la segunda.
- Lista de palabras que aparecen en la segunda lista, pero no en la primera.
- Lista de palabras que aparecen en ambas listas.

Nota: Para evitar las repeticiones, el programa deberá empezar eliminando los elementos repetidos en cada lista.

```

n = int(input("¿Cuántas palabras desea introducir en la primera lista? "))
l1 = []
for i in range(n):
    p = input("palabra "+str(i+1)+": ")
    l1.append(p)
print(l1)
for p in l1:
    while l1.count(p) > 1: # mientras la palabra esté repetida
        l1.remove(p)
print("Primera lista sin repeticiones:",l1)

n = int(input("¿Cuántas palabras desea introducir en la segunda lista? "))
l2 = []
for i in range(n):
    p = input("palabra "+str(i+1)+": ")
    l2.append(p)
print(l2)
for p in l2:
    while l2.count(p) > 1: # mientras la palabra esté repetida
        l2.remove(p)
print("Segunda lista sin repeticiones:",l2)

print("Lista de palabras que aparecen en cualquier lista:")
l = l1.copy()
for p in l2:
    if l1.count(p) == 0: # sólo si no está ya en la primera lista
        l.append(p) # se agrega a la lista unión
print(l)

print("Lista de palabras que aparecen en la primera lista, pero no en la segunda:")
l = []
for p in l1:
    if l2.count(p) == 0:
        l.append(p)
print(l)

print("Lista de palabras que aparecen en la segunda lista, pero no en la primera:")
l = []
for p in l2:
    if l1.count(p) == 0:
        l.append(p)
print(l)

print("Lista de palabras que aparecen en ambas listas:")
l = []
for p in l1:
    if l2.count(p) > 0:
        l.append(p)
print(l)
#####

```

9. Escriba un programa que permita crear una lista de palabras y que, a continuación, ordene la lista por orden alfabético.

```

n = int(input("¿Cuántas palabras desea introducir en la lista? "))
l = []
for i in range(n):
    p = input("palabra "+str(i+1)+": ")
    l.append(p)
print(l)
l.sort()
print("Lista ordenada alfabéticamente:", l)
#####

```



DESDE LA 1:00H
MARTES 31
DE DICIEMBRE

FIN DE AÑO 2024

THE END
FABRIK
- MADRID -

Valentina Otero Fernández

10. Escriba un programa que pida un número y a continuación escriba la lista de todos los divisores del número (incluidos el uno y él mismo)

```
n = int(input("Introduzca un número entero: "))
l = [1]
div = 2
while div <= n:
    if n % div == 0:
        l.append(div)
    div = div+1
print("los divisores de",n,"son:",l)
#####
#####
```

11. Escriba un programa que pida un número y a continuación escriba la lista de todos los números primos hasta él.

```
def es_primo(n):
    # devuelve True si el número es primo y False si es compuesto

    primo = True
    tope = n ** 0.5
    div = 2
    while primo and (div <= tope):
        if n % div == 0:      # n tiene al menos un divisor
            primo = False
        else:
            div = div + 1   # si no es divisor prueba con el siguiente
    return primo

n = int(input("Introduzca un número entero: "))
l = [1]
div = 2
while div <= n:
    if es_primo(div):
        l.append(div)
    div = div+1
print("los números primos hasta",n,"son:",l)
#####
#####
```

12. Escriba un programa que calcule términos de una sucesión del tipo $U_{n+1} = a \cdot U_n + b$. El programa tiene que pedir el valor de a, de b y del término U_0 y el número de términos a calcular.

```
a = int(input("Introduzca el coeficiente 'a': "))
b = int(input("Introduzca el coeficiente 'b': "))
u0 = int(input("Introduzca el término inicial 'U0': "))
n = int(input("Introduzca el número de términos a calcular: "))

l = [u0]
for i in range(1,n+1):
    l.append(a*l[i-1]+b)
print("La sucesión solicitada es:",l)
#####
#####
```

13. Escriba un programa que calcule términos de la sucesión $U_{n+1} = 3 \cdot U_n + 1$ si U_n es impar y $U_{n+1} = U_n/2$ si U_n es par. El programa tiene que pedir el término U_0 y el número de términos a calcular.

Nota: La conjectura de Collatz establece que esta sucesión alcanza el valor 1 para cualquier valor inicial. Desde 1937, esta conjectura permanece sin demostrar

```
u0 = int(input("Introduzca el término inicial 'U0': "))
n = int(input("Introduzca el número de términos a calcular: "))

l = [u0]
for i in range(1,n+1):
    if l[i-1]%2 == 0:      # si Un es par
        l.append(l[i-1]/2)
    else:      # si Un es impar
        l.append(3*l[i-1]+1)
print("La sucesión solicitada es:",l)
#####
#####
```

REGLA CLÁSICO

ENTRADAS
ANTICIPADAS
ESCANEANDO
ESTE QR.



14. Escriba un programa que solicite las dimensiones n y m de una matriz, sus elementos de uno en uno y la construya. Al finalizar de almacenar todos los elementos en la matriz, debe de mostrarla por pantalla.

```
...
f = int(input("Introduzca el número de filas de la matriz: "))
c = int(input("Introduzca el número de columnas de la matriz: "))

M = []
for i in range(1,f+1):
    fila = []
    for j in range(1,c+1):
        e = float(input("Introduzca el elemento en la fila "+str(i)+", columna "+str(j)+": "))
        fila.append(e)
    M.append(fila)

print(M)

for i in range(f):
    fila = ""
    for j in range(c):
        fila = fila + " "+str(M[i][j])
    print(fila)
```