

Tarea – Ciclos y funciones

1. Crea un programa que lea un número del teclado e imprima la suma de 0 a ese número.

```
main.py > ...
1 numero = int(input("Ingrese un número: "))
2 suma = 0
3
4 for i in range(1, numero+1):
5     suma += i
6
7 print("La suma de 0 a", numero, "es:", suma)
```

Ingrese un número: 10
La suma de 0 a 10 es: 55
>

2. Crea un programa que lea números del teclado indefinidamente hasta que el usuario introduzca el número 0. Una vez introducido el 0. Imprime el promedio de todos los números introducidos (incluyendo el 0).

```
main.py > ...
1 numeros = []
2 numero = int(input("Pon un número : "))
3
4 while numero != 0:
5     numeros.append(numero)
6     numero = int(input("Pon otro número: ") #si pones cero se termina
7
8 promedio = sum(numeros) / len(numeros)
9
10 print("promedio de los números es:", promedio)
```

Pon un número : 34
Pon otro número: 56
Pon otro número: 76
Pon otro número: 34
Pon otro número: 45
Pon otro número: 56
Pon otro número: 67
Pon otro número: 89
Pon otro número: 34
Pon otro número: 56
Pon otro número: 0
promedio de los números es: 54.7
>

3. Crea un programa que cree una lista de compras. Primero le pide al usuario que ingrese la cantidad n de artículos. Luego leen artículos de la terminal. Una vez leídos todos, imprime la lista en orden alfabético.

4. Crea un programa filtrador de números pares. Primero pide al usuario que ingrese un número n, la cantidad de números en la lista. Luego filtra la lista de números para que solo tenga números pares e imprime la lista.

```
main.py > ...
1 numeros = []
2 cantidad = int(input("Cantidad de números en la lista: "))
3
4 for i in range(cantidad):
5     numero = int(input("Pon un número: "))
6     numeros.append(numero)
7
8 numeros_pares = [num for num in numeros if num % 2 == 0]
9
10 print("Los números pares son:")
11 for num in numeros_pares:
12     print(num)
```

Cantidad de números en la lista: 6
Pon un número: 1
Pon un número: 2
Pon un número: 3
Pon un número: 4
Pon un número: 52
Pon un número: 34
Los números pares son:
2
4
52
34
>

5. Recibe un String de entrada e imprime sólo las vocales:

Por ejemplo “Carlos, quien trabaja en Google, es un muy buen programador” Daría como resultado: “aouieaaaaeoeeuuueoaa”

```

main.py > ...
1 cadena = input("Pon texto: ")
2 vocales = ""
3
4 for letra in cadena:
5     if letra.lower() in "aeiouáéíóú":
6         vocales += letra
7
8 print("Vocales que tiene tu oracion:")
9 print(vocales)
10

```

Pon texto: danielita se fue al mercadito a comprar panesitos para la abuelita
Vocales que tiene tu oracion:
aieiaevaeaaioaooaaioaaaaeiea

6. Crea una función que reciba un número y regrese un booleano indicando si es divisible entre 243

```

main.py > ...
1 def divisiblepor243(numero):
2     if numero % 243 == 0:
3         return True
4     else:
5         return False
6
7 numero = int(input("Pon un número: "))
8
9 if divisiblepor243(numero):
10    print("Divisible por 243")
11 else:
12    print("No es divisible por 243")

```

Pon un número: 3479238745
No es divisible por 243

7. Crea una función llamada multiplicarString que reciba un string y un número entero positivo. La función debe regresar el mismo string repetido n veces. Por ejemplo:

“Hola” y 3 → “HolaHolaHola”

“Adios” y 5 → “AdiosAdiosAdiosAdiosAdios”

```

main.py > ...
1 def texto(string, n):
2     if n <= 0:
3         return ""
4     else:
5         return string * n
6
7 string = input("Pon string: ")
8 n = int(input("Numero de repiticiones: "))
9
10 resultado = texto(string, n)
11 print(resultado)

```

Pon string: HOLA
Numero de repiticiones: 4
HOLAHOLAHOLAHOLA

8. Crea una función que reciba 3 números como parámetros (no en arreglo) y regrese la suma si al menos uno de esos números es mayor que 100, si no regresa la multiplicación de los 3.

9. Crea una función llamada `stringsExclusivos` que reciba 2 strings y regresa `true` si uno tiene puras vocales y el otro puras consonantes, regresa `false` en cualquier otro caso. Ejemplo:

“aeaei” , “qwt” → `True`

“aaec” , “ttsdf” → `False`

10. **Crea una función que se llame `esPalindromo` que reciba un string cualquiera y regrese `true` si es palíndromo, y `false` en el caso contrario.**

Ejemplo:

“hola” → **`False`**

“kayak” → **`True`**



```

main.py > ...
1 def esPalindromo(string):
2     string = string.lower()
3     string = string.replace(" ", "")
4     reverse_string = string[::-1]
5
6     if string == reverse_string:
7         return True
8     else:
9         return False
10
11 cadena = input("Texto: ")
12
13 resultado = esPalindromo(cadena)
14 print(resultado)

```

Texto: kayak
True

11. **Challenge:** Crea una función llamada `aBinario` que reciba un número entero positivo y regrese un arreglo de ceros y unos que sea la representación en el sistema numérico binario del número recibido. Si no estás familiarizado con el sistema binario, dale primero una leída a este artículo: <https://www.elvisualista.com/2016/10/20/que-son-los-numeros-binarios/>

Ejemplos:

$5 \rightarrow [1, 0, 1]$

$2 \rightarrow [1, 0]$

$1 \rightarrow [1]$

$45 \rightarrow [1, 0, 1, 1, 0, 1]$

12. (Challenge)

Imprime una pirámide como la siguiente:

1

1 2

1 2 3

1 2 3 4

1 2 3 4 5

Vas a recibir de entrada un número n que indica la cantidad de filas que debes imprimir, y cada fila incrementa su longitud en 1 como se ve en el ejemplo.