CARRERA DE: INGENIERÍA INFORMÁTICA MATERIA: PARADIGMA DE PROGRAMACIÓN FORMACIÓN TEÓRICA: GUÍA DE EJERCICIOS

## GUÍA DE EJERCICIOS

## **TEMA: EJERCICIOS GENERALES EN PYTHON**

OBJETIVO: Familiarizarse con el lenguaje utilizando las estructuras mas comunes del lenguaje. Mostrar su utilidad y comprobar su funcionamiento.

- 1. Escribir una función que reciba como parámetros el inicio y fin (inclusive) de un rango numérico. La función debe:
  - a. Imprimir en pantalla todos aquellos números que sean divisibles por 7 pero no sean divisibles por 5.
  - b. Imprimir el mismo resultado anterior, pero separados por coma.

Resultado Esperado: Por ejemplo, si se invoca con los parámetros 1 y 100 (puntos a y b)

```
14
21
28
42
49
56
63
77
84
91
98
```

- 2. Escribir la función factorial, que reciba como parámetro el numero inicial y compute su resultado.
  - a. Ejemplo factorial(8) = 8\*7\*6\*5\*4\*3\*2\*1 = 40320. Recuerde que factorial de 0 por definición es 1.
  - b. Hacer la implementación inversa (si lo hizo recursivo, hacerlo iterativo o viceversa)
- 3. Escribir una función que genere y retorne un diccionario ASCII. Para ello, las claves deben ser las letras a partir de la 'a' y el valor debe ser el número ASCII (a -> 97, b -> 98, ...).

Tips: se puede utilizar la función **chr** para convertir un número en su correspondiente letra o **ord** para la situación inversa (conocer el valor ASCII de una letra). También recordar que se puede crear un diccionario vacío con **dict()**.

## Resultado Esperado:

```
{'a': 97, 'b': 98, 'c': 99, 'd': 100, 'e': 101, 'f': 102, 'g': 103, 'h': 104, 'i': 105, 'j': 106, 'k': 107, 'l': 108, 'm': 109, 'n': 110, 'o': 111, 'p': 112, 'q': 113, 'r': 114, 's': 115, 't': 116, 'u': 117, 'v': 118, 'w': 119, 'x': 120, 'y': 121, 'z': 122}
```

4. Escribir un programa que reciba como parámetro un string de elementos separados por coma y retorne una lista conteniendo cada elemento.

Resultado Esperado: Si se utiliza como parámetro el string '14, Juana Perez, True' se espera que la función retorne la lista ['14', 'Juana Perez', 'True']

b. Modificar la función anterior, pero obteniendo correctamente cada dato según su tipo (numero entero, string o boolean). Se considera que el string siempre tiene la forma de primer campo un entero, luego un texto y por ultimo un boolean.

Resultado Esperado: Si se envia '14, Juana Perez, True' se espera que la función retorne la lista [14, 'Juana Perez', True]

c. Utilizando la función realizada en el punto 4b, hace una nueva función que reciba una lista de strings y retorna otra lista, como listas con los campos formateados (mantener el mismo formato de string que el punto b, es decir, entero, texto y boolean).

Resultado Esperado: Si se envía la siguiente lista como parámetro de la función:

```
[
'14,Juana Perez,True',
'16,Raul Dell,False',
'18,Mariana Castillo,True',
'176,Pedro Rodríguez,False'
```

## Se espera obtener la lista:

```
[[14, 'Juana Perez', True], [16, 'Raul Dell', True], [18, 'Mariana Castillo', True], [176, 'Pedro Rodríguez', True]]
```

5. Escriba una función que reciba como parámetro el radio de un circulo y devuelva una tupla conteniendo en el primer elemento el perímetro y en el segundo el área del mismo.

<u>TIPs</u>: Recuerde que la fórmula del perímetro es (2 \* pi \* r) y el área se define como  $(pi * r^2)$ . Se puede utilizar la constante pi definida en el módulo math (import math)

Resultado Esperado: Si se utiliza la funcion con el valor de r=5, entonces debe devolver la tupla (31.415, 78.539)

6. Se necesita un programa que reciba por línea de comando una serie de palabras, hasta que reciba la palabra "exit". Una vez recibida dicha instrucción, debe mostrar por salida standard las mismas palabras ingresadas, almacenadas en una lista, pero ordenadas alfabéticamente y cada una debe estar capitalizada.

Resultado Esperado: El programa le solicita al usuario que ingrese y este escribe:

```
Ingrese palabra: hola
Ingrese palabra: QUE
Ingrese palabra: tal
Ingrese palabra: como
```

```
Ingrese palabra: estas
Ingrese palabra: exit
```

```
Se espera recibir el siguiente resultado: ['Como', 'Estas', 'Hola', 'Que', 'Tal']
```

7. Se necesita una función que reciba un texto de varias palabras separadas por un espacio, con letras mayúsculas y minúsculas, y retorne una colección las palabras utilizadas, todas en minúscula y sin duplicados

Resultado Esperado: Si por ejemplo el programa solicitara ingresar el texto y el usuario ingresa:

Por favor, ingrese un texto: HOLA NO Deberia deberia haber duplicados de de de de ningun tipo

```
Se debería obtener como resultado: {'haber', 'ningun', 'duplicados', 'deberia',
'tipo', 'no', 'de', 'hola'}
```

8. En Años anteriores, se necesitaba una función en python que reciba un texto conteniendo bits (simbolos 1 y 0), y debia armar una lista conteniendo 8 bits por elementos (1 byte). Por ejemplo, si se incova la funcion con el siguiente texto como parámetro:

El programador de ese momento armó el siguiente código:

```
def ej08a(texto):
    """Arma una lista de bytes acorde al texto recibido por parametro"""
    indice = 0
    resultado = []
    current_byte = ""

for i in texto:
        current_byte += i  # se agrega el nuevo caracter al byte actual
        indice += 1  # se incrementa en uno el indice

    if indice % 8 == 0:
        # Comienza un nuevo byte
        resultado.append(str(current_byte))
        current_byte = ""

return resultado
```

que funciona según lo estipulado (probarlo para entenderlo). Se pide que analice el código y que conteste lo siguiente:

- a. ¿Qué cambios agregaría a la función para verificar que el texto ingresado solo contiene caracteres "1" o "0" antes de ser procesado?
- b. ¿Qué objetivo tiene la validacion indice % 8 == 0?
- c. ¿En la línea 13, porque usa la función **str** si **current\_byte** ya es un string?
- d. ¿Existe alguna manera rápida y sencilla si se pide modificar la función para que la lista resultado no contenga duplicados? En caso de existir, explique cuál sería.

9. Se necesita un programa que reciba por parámetro un texto, y que devuelva una tupla conteniendo en el primer lugar, la cantidad de letras (mayúsculas o minúsculas), en el segundo lugar la cantidad de dígitos numéricos y en el tercer lugar, otros símbolos.

Resultado Esperado (tupla): Por ejemplo, si se utiliza como parámetro el texto: "Esta es una mañana LLuviosa!! 25 días más serán así??" se debería obtener como resultado la tupla (38, 2, 13)

<u>TIPs</u>: Python cuenta con las funciones isdigit() y isalpha() utilizadas sobre strings (o "chars") que pueden ser útiles para identificar tipos de caracteres. Por ejemplo, '9'.isdigit() retorna True y '!'.isalpha() retorna False. Buscar en la documentación mayor detalle.