

Trabajo Práctico 3

1) Ejercicio de ingeniería inversa.

Como examen para el ingreso a una empresa te piden que mires un código que tiene algunos errores y que no se sabe qué hace o qué intenta hacer. Tu misión consiste en contestar / resolver:

- ¿Qué hace o qué pretende hacer el código?
- ¿Qué errores tiene? Enumerarlos. En algún momento, realizar un seguimiento con algún debug. Incluir una o más capturas de pantalla señalando en dónde detectaron alguno de los errores. (Ayuda: antes deberán hacer alguna corrección al código para que este seguimiento tenga sentido).
- ¿Qué cosas señalarías que no son errores, pero no se corresponden con las buenas prácticas explicadas en clase? Enumerarlas.
- Corregir el programa para que funcione correctamente y cumpla con las buenas prácticas enseñadas.

```
p = [5, 3, 8, 9, 7]
def calc(a, b, c):
    if (c == 0):
        return -1
    elif (a[c] == b):
        return c
    else:
        calc(a, b, c - 1)
```

```
x = input()
calc(p, x, length(p))
```

Puntos 15

2) Escribir una función *validar* que reciba una cadena de caracteres y devuelva True si una clave contiene las restricciones requeridas, False de lo contrario.

- Debe contener entre 8 y 12 caracteres
- Debe contener por lo menos un carácter alfabético en mayúsculas (sin acentuar)
- Debe contener por lo menos un carácter alfabético en minúsculas (sin acentuar)
- Debe contener por lo menos un dígito numérico
- Debe contener por lo menos alguno de los siguientes símbolos: [*, -, \$, @]
- Cualquier otro símbolo que no esté en el grupo indicado rechazará la validación

Ejemplos:

Llamado	Devolución	Observaciones
validar("Algoritmo \$1")	False	Entre la 'o' y el '\$' hay un espacio en blanco
validar("Aprobé-con-7")	False	La é está acentuada
validar("Algoritmo\$1")	True	
validar("Aprobe-con-7")	True	

Importar el módulo doctest e imprimir el resultado de la ejecución de las tres pruebas que figuran en el enunciado más otras dos pruebas más.

Puntos 20

- 3) En el programa “Camino a la fama”, los participantes muestran sus aptitudes: baile, canto, etc. Un jurado vota colocando un puntaje del 1 al 10 a los participantes. Esto genera una lista de listas (cada sublista es una votación). Ejemplo: [[“Luisa”, 4], [“Mariano”, 10], [“Luisa”, 5], etc...]. Las votaciones son de distintos jueces y por distintas actuaciones por lo que los participantes aparecerán varias veces.

Se pide que escribas un programa en Python que:

- Procese esa lista y genere un diccionario con clave *participante* y valores: *sumatoria_puntaje*, *cantidad_puntajes*, *promedio_puntaje*. Los valores estarán dentro de una lista.
- Luego, debe listar: participantes – promedio_puntaje, ordenados de mayor a menor por promedio_puntaje. Con el ejemplo anterior debería quedar:

```
Mariano 10
Luisa    4.5
```

En un main armar una lista de ejemplo y verificar el resultado.

Puntos 25

Consideraciones: en todos los ejercicios pedimos que cumplan con las prácticas que explicitamos en el curso (nombres, estilos, etc.) y prestar especial atención a la modularización (ejercicio 3) y la eficiencia (no hacer ciclos de más, etc.).

- El ejercicio 1 deben entregarlo en formato pdf.
- Los ejercicios 2 y 3 en archivos .py por separado.
- Los nombres de los archivos deben ser: apellido_e3_# (reemplazar # por 1, 2 o 3).
- Finalmente, los 3 archivos los entregan en un único archivo comprimido con nombre apellido_e3.

Fecha límite de entrega: domingo 11 de febrero a las 23.59hs.