























EJEMPLO DE PARCIAL 1

ANÁLISIS DE CÓDIGO

Analizar el siguiente código y responder a las preguntas de abajo, circulando  para verdadero, o  para falso:

```
1 # Función
2 def analizarLista(miLista, valor = 0):
3
4     if valor in miLista:
5         print("El valor", valor, "está en la lista.")
6     else:
7         print("El valor", valor, "no está en la lista.")
8
9     sublista = miLista[len(miLista) // 2::]
10
11     mayoresQueValor = [elemento for elemento in sublista if elemento > valor]
12
13     # Imprimir información para análisis
14     print("Lista original:", miLista)
15     print("Sublista aleatoria seleccionada:", sublista)
16     print("Elementos destacados", mayoresQueValor)
17
18     return
19
20 # Ejemplo de uso de la función
21 listaEjemplo = [12, 1, 22, 14, 6, 32, 18, 5, 0, 20]
22 analizarLista(listaEjemplo)
23 print()
24 analizarLista(listaEjemplo, 14)
25
```

1. La llamada a la función **analizarLista** de la línea 22 arrojará error por falta de parámetro.  
2. La lista **miLista** es modificada dentro de la función.  
3. La función **analizarLista** incluye dentro de su código una lista por comprensión.  
4. Al ejecutarse la línea 9, la lista **sublista** pasará a tener los últimos 5 elementos de la lista **miLista**.  
5. En la función **analizarLista** hay variables locales.  
6. La instrucción **return** de la función **analizarLista** está mal indentada.  
7. El nombre **valor** es un nombre válido para un parámetro de función.  
8. El nombre de la función de la línea 2 no respeta el estilo de nombres camelCase.  
9. El comentario de la línea 1 es un comentario docstring de la función **analizarLista**.  
10. El print de línea 16 mostrará **Elementos destacados [32, 18, 20]** cuando la función sea llamada desde línea 24.  

EJERCICIO POR DESARROLLAR

Se necesita desarrollar un programa para simular un juego de ruleta. Ya se cuenta con una plantilla con la estructura base del programa:

```
-----  
MENÚ DEL SISTEMA  
-----  
[1] Generar lanzamientos simulados de ruleta  
[2] Informar las veces que ganó la banca  
[3] Informar en qué lanzamientos ganó la banca  
[4] Mostrar otras estadísticas  
-----  
[0] Salir del programa  
  
Seleccione una opción: |
```

Se te proveerá de un archivo .py con la estructura del menú armada, en donde deberás desarrollar el código de las funciones que se solicitan a continuación (en el código, aparecen tres puntos ... en todos los lugares donde se espera que codifiques). Tené en cuenta que el programa incluye algunos comentarios de ayuda y además se compone de varios bloques:

- 1- Bloque de importación: Si corresponde, aquí deberás agregar las importaciones necesarias.
- 2- Bloque de funciones: Esta será la sección en donde desarrollarás el código para cada función.
- 3- Bloque de inicialización de variables: Si corresponde, aquí deberás declarar las variables y listas que necesites. Deberás respetar el nombre de la lista que ya está codificada y que se llama **lanzamientos**.
- 4- Bloque de menú: Aquí deberás llamar a las correspondientes funciones y hacer las impresiones de resultados. EN ESTA OPORTUNIDAD SE PIDE: Ninguna función incluirá instrucciones print(...).

[1] Generar lanzamientos simulados de ruleta

Al seleccionar esta opción, dentro de la función el usuario podrá cargar la cantidad de lanzamientos de ruleta que se simularán. **Deberás validar que ingrese un valor entre 10 y 30 lanzamientos, con pedido de reintento en caso de error.**

Una vez ingresada una cantidad de N lanzamientos, la función generará una lista de N elementos con números al azar entre 0 y 36 (valores de la ruleta) y devolverá la misma al programa principal. En el programa principal se imprimirán los valores generados.

Se verá así (ajustar los mensajes y las impresiones de acuerdo con esta imagen):

```
Seleccione una opción: 1  
  
Ingrese la cantidad de lanzamientos [de 10 a 30]: 5  
El valor ingresado es inválido, vuelva a intentar [de 10 a 30]: 40  
El valor ingresado es inválido, vuelva a intentar [de 10 a 30]: 25  
Lanzamientos:  
[27, 6, 18, 27, 0, 26, 27, 11, 11, 1, 32, 8, 0, 4, 33, 7, 28, 0, 24, 28, 19, 2, 14, 19, 36]  
  
Presione ENTER para volver al menú.|
```

[2] Informar las veces que ganó la banca

Al seleccionar esta opción, dentro de la función deberás contar la cantidad de veces que apareció el número 0 en la lista de lanzamientos generada en el punto 1. La función devolverá dicha cuenta y deberás imprimir el resultado en el programa principal.

EN ESTA OPORTUNIDAD SE PIDE: No utilices en esta función el método count().

Se verá así (para el caso de los datos de la imagen del punto 1) (ajustar los mensajes y las impresiones de acuerdo con esta imagen):

```
Seleccione una opción: 2

La banca gana 3 veces.

Presione ENTER para volver al menú.
```

[3] Informar en qué lanzamientos ganó la banca

Al seleccionar esta opción, la función deberá determinar en qué lanzamientos salió el número 0. La función devolverá una lista con los números de lanzamientos y deberás imprimir el resultado en el programa principal.

Se verá así (para el caso de los datos de la imagen del punto 1) (ajustar los mensajes y las impresiones de acuerdo con esta imagen):

```
Seleccione una opción: 3

La banca gana en los lanzamientos: [5, 13, 18]

Presione ENTER para volver al menú.
```

En el caso que en ningún lanzamiento gane la banca, el mensaje será el de la siguiente imagen:

```
Seleccione una opción: 3

La banca no gana en ningún lanzamiento.

Presione ENTER para volver al menú.
```

[4] Mostrar otras estadísticas

Al seleccionar esta opción, la función deberá determinar:

- Cuántos fueron los números pares que salieron, y cuáles fueron.
- Cuántos fueron los números impares que salieron, y cuáles fueron.
- Cuál fue el promedio de los números que salieron.

La función devolverá 5 datos de salida, que, según los cálculos anteriores algunos serán enteros y otros serán listas.

EN ESTA OPORTUNIDAD SE PIDE: Siempre que sea posible, dentro de la función usar listas por comprensión para resolver cada cálculo.

EN ESTA OPORTUNIDAD SE PIDE: En la llamada a la función, dentro del programa principal, deberás usar la técnica de desempaqueado, es decir que lo devuelto por la función sea desempaqueado en una única línea en 5 variables globales.

Se verá así (para el caso de los datos de la imagen del punto 1) (ajustar los mensajes y las impresiones de acuerdo con esta imagen):

```
Seleccione una opción: 4

lanzamientos pares: 15 [6, 18, 0, 26, 32, 8, 0, 4, 28, 0, 24, 28, 2, 14, 36]
lanzamientos impares: 10 [27, 27, 27, 11, 11, 1, 33, 7, 19, 19]
Valor promedio: 16.32

Presione ENTER para volver al menú.
```