

Laboratorio (Semana 4)

(5 puntos)

1. Descripción de la Actividad

Se quiere que realice programas en Python que resuelvan los siguientes problemas. Todas las soluciones deberán estar en un solo script `LabSem4.py`. No serán necesarias verificaciones del input, es decir, si el requerimiento indica que se recibirá un número entero positivo, entonces solo se probarán números enteros positivos.

1. `obtener_subcadena`: Dadas dos cadenas de caracteres `s` y `A`, se quiere obtener la posición de la primera instancia de `s` en `A`. Si `s` no es una sub-cadena de `A`, retornar `-1`.

Ejemplos de ejecución:

```
es_subcadena("world", "Hello world!") # 6
es_subcadena("world?", "Hello world!") # -1
```

2. `contar_ocurrencias`: Dadas dos cadenas de caracteres, se quiere contar cuantas veces aparece la primera cadena en la segunda.

Ejemplos de ejecución:

```
contar_ocurrencias("world", "Hello world!") # 1
contar_ocurrencias("world?", "Hello world!") # 0
contar_ocurrencias("aa", "aaa") # 2
```

3. `comparar_fechas`: Dadas dos fechas, se quiere saber si la primera es mayor que la segunda. Las fechas vienen en formato `"DD-MM-YYYY"`.

Hints:

- Puedes obtener sub-strings usando la misma sintaxis `[:]` para las listas.
- Puedes convertir strings en enteros usando `int`, por ejemplo `int("0")`. Pero generará un error si el string no corresponde a un entero.

Ejemplos de ejecución:

```
comparar_fechas("05-05-2002", "01-01-2000") # True
comparar_fechas("05-05-1999", "01-01-2000") # False
```

4. `ordenado`: Dada una lista de flotantes y un booleano `B`, se quiere verificar si la lista está ordenada ascendentemente si `B` es `True`, o si la lista está ordenada descendentemente si `B` es `False`.

Ejemplos de ejecución:

```
ordenado([0, 1, 2, 3], True) # True
ordenado([3, 2, 1, 0], True) # False
ordenado([3, 2, 1, 0], False) # True
```

5. `es_permutacion`: Dada una lista y un entero N , se quiere saber si la lista es una permutación de la secuencia $\langle 0, 1, 2, \dots, N-1 \rangle$.

Ejemplos de ejecución:

```
es_permutacion([0, 1, 2, 3], 4)    # True
es_permutacion([1, 2, 3, 4], 4)    # False
es_permutacion([0, 1, 2, 3], 3)    # False
```

6. `suma_maxima`: Dado un arreglo de números enteros, se quiere encontrar el segmento $[p..q]$ tal que su suma sea la máxima posible. El método debe retornar el valor de la suma.

Ejemplos de ejecución:

```
suma_maxima([4, -1, 4, -8, 5, 1])  # 7
suma_maxima([-1, -8, 3, 2, -6, 4]) # 5
```

7. `desviacion_estandar`: Dada una lista de flotantes, se quiere calcular la desviación estándar de los elementos de la misma. El cálculo de la desviación estándar debe realizar mediante la fórmula $\sqrt{\frac{1}{N} \sum_{i=0}^N (S[i] - \mu)^2}$, donde S , N y μ son la lista, la longitud de la lista y el promedio de la lista respectivamente.

Ejemplos de ejecución:

```
desviacion_estandar([1, 1, 1, 1, 1, 1]) # 0.0
desviacion_estandar([1, 1, 1, 2, 2, 2]) # 0.5
```

8. `diferencia_matriz`: Dada una matriz cuadrada M de flotantes y un entero N , se quiere calcular la diferencia entre la suma de los elementos de la sub-matriz $N \times N$ que se encuentra en la esquina superior izquierda de M , y la suma del resto de los elementos.

Ejemplos de ejecución:

```
matriz: List[List[float]] = [
    [1.0, 1.0, 2.0],
    [1.0, 1.0, 2.0],
    [2.0, 2.0, 2.0],
]
diferencia_matriz(matriz, 2) # -6.0
diferencia_matriz(matriz, 3) # 14.0
```

9. `logaritmo`: Dado dos números enteros positivos R y B , se quiere encontrar el mayor número entero P tal que $B^P \leq R$. El valor por defecto de B debe ser 2. No se pueden usar funciones que calculen logaritmos.

Ejemplos de ejecución:

```
logaritmo(16) # 4
logaritmo(18, 10) # 1
```

10. `cuadrado_magico`: Dada una matriz cuadrada de enteros, determina si es un cuadrado mágico. Un cuadrado mágico es una matriz cuadrada de números enteros, tal que las sumas de los números en cada fila, cada columna y las dos diagonales principales son iguales.

Ejemplos de ejecución:

```
matriz: List[List[int]] = [
    [2, 7, 6],
    [9, 5, 1],
    [4, 3, 8],
]
cuadrado_magico(matriz) # True
matriz = [
    [2, 7, 1],
    [9, 5, 1],
    [4, 3, 8],
]
cuadrado_magico(matriz) # False
```

2. Evaluación

Se implementará un script para la evaluación de cada solución propuesta, el cual aplicará una variedad de casos de prueba, y generará un resultado final. Se evaluarán las firmas de las funciones. Aunque no se realizará una evaluación del estilo de codificación, se efectuarán correcciones en este aspecto, lo cual les será de utilidad en evaluaciones futuras, donde dicho estilo podría ser objeto de evaluación.

3. Condiciones de Entrega

El programa correspondiente al laboratorio `LabSem4.py` y la declaración de autenticidad debidamente firmada, deben ser incluidos en un archivo comprimido, en formato `tar.xz` (o `zip`), denominado `LabSem4_X.tar.xz` (o `LabSem4_X.zip`), donde `X` representa el número de carné del estudiante. La entrega del archivo debe realizarse a través de Gmail, enviándolo al correo electrónico **16-10072@usb.ve** antes de las 11:59 PM del día domingo 11 de febrero de 2024. Las entregas que se realicen después de la hora límite tendrán una penalización de un (1) punto, más un punto adicional por cada hora de retraso. Por lo tanto, después de 4 horas de retraso, la evaluación será anulada.