

IWI131 Programación (Fase 2) — Ejercicios Semana 1: Strings

1. Uno de los grupos de música más importantes de Pythonia es Deuman. Sus integrantes han producido un nuevo disco corto de cuatro canciones, las que llamaremos: A, B, C y D. El grupo quiere saber cuál de las canciones tiene mejor recepción por parte del público y para ello realizará una encuesta a 1000 personas. Cada persona debe ingresar un dígito entre 1 y 7 para indicar cuánto le gustó cada canción: 1 es lo peor y 7 es lo máximo. Por ejemplo, si una persona ingresa 3675, significa que le asignó nota 3 a la canción A, un 6 a la canción B, 7 a la canción C y 5 a la canción D. Para hacer el proceso más automático, Deuman le solicita a Usted implementar lo siguiente:

a) Escriba la función `mejor(votos)`, donde `votos` es un entero de 4 dígitos que almacena los votos de las canciones, en la forma descrita. Los votos contendrán siempre 4 dígitos entre 1 y 7. La función debe retornar la canción mejor evaluada, es decir, el string `'A'`, `'B'`, `'C'` o `'D'` dependiendo de cuál obtuvo la nota más alta. En caso de haber empate en la nota máxima, la función debe retornar cualquiera de las canciones que empataron en el máximo. Sin embargo, hay una excepción: si en el voto ninguna canción obtiene nota ≥ 4 , entonces la función debe retornar el string vacío: `''`, pues suponemos que a la persona encuestada en realidad no le gustó mucho ninguna canción. Observe que la función debe retornar su resultado, pero no imprimir nada.

Ejemplo:

```
>>> print ( mejor ( 1621 ) )
'B'
>>> print ( mejor ( 1166 ) )
'C'
>>> print ( mejor ( 1223 ) )
''
```

b) Utilizando la función anterior, se le pide ahora que escriba un programa que pregunte su voto a las 1000 personas encuestadas. El programa debe responder cuál fue la canción que resultó favorita en la mayor cantidad de votos. En caso de empate, puede escribir cualquiera de las que empataron en el máximo. Suponga que los votos contendrán siempre 4 dígitos entre 1 y 7.

Ejemplo:

```
Voto ? 6727
Voto ? 1166
Voto ? 7117
...
Voto ? 7777
La favorita es D con 510 votos.
```

Nota: Usted puede contestar a esta pregunta aunque no haya respondido la parte (a) o aunque la tenga incorrecta. Lo importante es que utilice adecuadamente la función tal y como se describió, haciendo los llamados de manera adecuada, en el momento oportuno y utilizando correctamente lo que la función retorna.

2. Un grupo de bio-tecnólogos externos a la USM, tienen demasiado trabajo analizando cadenas de ADN. Todo el trabajo lo realizan a mano ya que no tuvieron un buen ramo de programación. Por esto, le piden a los estudiantes del ramo de Programación IWI-131 que les ayuden. Las cadenas de ADN están compuestas por 4 bases nitrogenadas (A: Adenina, C: Citosina, G: Guanina y T: Timina) agrupadas en bloques de 4. Ahora usted debe ayudar en las siguientes operaciones:

- a) Escriba la función `valida(cadena)` que reciba un string con una cadena de ADN y retorne **True** si la cadena es válida o **False** si no lo es. Una cadena no es válida cuando aparecen bases nitrogenadas distintas a las antes descritas.

```
>>> valida('CTGA CTGA AATT GGGC CTGG CCCC')
True
>>> valida('CTGA XCGA CGAT GGTA ACCC CCPC TTAA')
False
```

- b) Escriba la función `cantidad(cadena, base)` que reciba un string con una cadena de ADN y una base nitrogenada. La función debe retornar la cantidad de apariciones de la base en la cadena.

Nota: no puede utilizar el método `count` de procesamiento de texto.

```
>>> cantidad('CTGA CTGA AATT GGGC CTGG CCCC', 'A')
4
```

- c) Los científicos encontraron un patrón de clasificación, que se deduce de la cantidad mayoritaria de un cierto par de bases. Si la suma de las cantidades de Citosina y Guanina es mayor a la de Adenina y Timina, es una especie vegetal, en caso contrario es una especie animal. Escriba un programa que pregunte la cantidad de cadenas de ADN a evaluar, luego solicite las cadenas y finalmente muestre las cantidades de cada especie y cadenas no válidas.

```
Cantidad de cadenas de ADN: 3
Ingrese cadena 1: CGTA CAGT TTGG GGTA AATG CATG
Ingrese cadena 2: CACC CTGA GGAA ACAA XTFC ATGG
Ingrese cadena 3: TGTG TTGA ATGA CTAT ATTT
Cantidad animales: 2
Cantidad vegetales: 0
Cantidad no validas: 1
```

3. La empresa de seguridad SecurityUSM ha solicitado a los alumnos de programación realizar las siguientes funciones y un programa que las utilice para completar su nueva propuesta de seguridad.

a) Escriba una función `ordenado(n)` que verifique si los dígitos de un número entero positivo están ordenados de mayor a menor.

```
>>> ordenado(567)
False
```

```
>>> ordenado(9876)
True
```

```
>>> ordenado(98876)
True
```

b) Escriba una función `crear_password(numero)` que reciba un número entero positivo y retorne la password asociada según la siguiente regla: encontrar el mayor y el menor número de 2 dígitos (seguidos) dentro de `numero` y luego juntar el mayor con el menor para formar la password (4 dígitos). Asuma que `numero` tendrá al menos 2 dígitos.

```
>>> crear_password(356725512199)
9912
356725512199
>>> crear_password(821012)
8201
821012
>>> crear_password(123)
2312
123
```

c) Realice un programa que pida números al usuario. Por cada número ingresado el programa debe generar una password (no mostrarla). Cuando el usuario escriba `FIN`, el programa debe mostrar: cuantas passwords fueron generadas, cuantas de ellas están ordenadas de mayor a menor y cuantas no. Asuma que el usuario solo ingresará números de dos o más dígitos, o bien la palabra `FIN`.

```
Numero: 389566712
Numero: 821012
Numero: 311111
Numero: 567123
Numero: FIN
Hay 4 passwords
1 ordenadas
3 no ordenadas
```

```
Numero: 389516721
Numero: 117123
Numero: 9876588654
Numero: FIN
Hay 3 passwords
2 ordenadas
1 no ordenadas
```

4. Una de las nuevas formas de expresión respecto del sentir ciudadano son las redes sociales, en las cuales los usuarios pueden entregar su opinión respecto de diversos temas. La red social más popular es Pytweet. El servicio de estadísticas requiere recuperar la información asociada a los mensajes (Ptweets) relacionados con el pensamiento de la gente en torno a la situación económica del país. Una particularidad de los Ptweets es que estos contienen caracteres especiales que no pertenecen al lenguaje español, los cuales corresponden a: @, {, }, - .

Desarrolle un programa en Python que permita leer uno a uno los Ptweets de la gente hasta que se ingrese la palabra "Finish". El programa deberá mostrar el siguiente resumen: cantidad de Ptweets, total de caracteres ingresados, total de Ptweets los cuales tienen al menos un carácter especial, frecuencia de aparición de cada carácter especial, donde la frecuencia de aparición se calcula como el número de apariciones del carácter dividido por el total de caracteres leídos. Un ejemplo de la ejecución de este programa se muestran a continuación:

```
Ingrese Ptweet: @Robin123: la economia viene en baja {@Charly2312_}
Ingrese Ptweet: @Charly2312 nada que hacer
Ingrese Ptweet: {saludos a todos} @allyou123
Ingrese Ptweet: Todos sabemos que pasaria
Ingrese Ptweet: @Speed23451: _como mejorar esto es el tema_
Ingrese Ptweet: Finish
```

```
Total tweets: 5
Total caracteres: 174
Total Ptweets con caracteres especiales: 4
Total @: 0.0287
Total {: 0.0115
Total }: 0.0115
Total _: 0.0172
```

Nota: considere que la frecuencia de aparición es un valor flotante de 4 cifras decimales.