

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

1. El ministro de educación de Pythonia tiene una política para atender las necesidades que tienen los colegios que están bajo su jurisdicción. Cada vez que se requiere hacer reparaciones en la infraestructura en un colegio, el ministro les sugiere organizar un bingo para que se puedan recaudar los fondos necesarios. Para jugar al bingo, se utilizan cartones que contienen N números **distintos** y **desordenados**, representados por strings de la forma `'numero1-numero2-...-numeroN'` con el caracter guión `'-'` como separador de los números. Tener en cuenta que los números x del cartón están en el rango $0 < x \leq 99$, es decir, tienen solo 2 dígitos.

En base a lo anterior, se le solicita:

1. Escriba la función `verificar_numero(numero, carton)` que recibe como parámetro un número entero y un string con un carton de bingo. Esta función deberá retornar `True` si el número ingresado está contenido en el cartón de bingo y `False` en caso contrario.

```
>>> verificar_numero(2, '1-12-54-5')
False
>>> verificar_numero(54, '1-12-54-5')
True
```

2. Escriba un programa que simule un juego de bingo para dos jugadores y muestre al ganador. Deberá ingresar como entrada los cartones de cada jugador y luego los números que vayan saliendo en la tómbola. El ganador será aquel jugador que logre marcar todos los números de su cartón. Si ocurre un empate deberá indicar que ambos jugadores ganaron.

Observaciones: Los datos de la tómbola que se van ingresando serán siempre correctos y no se debe considerar mal uso del programa, como por ejemplo, ingresar números repetidos de la tómbola.

```
Carton del Jugador 1: 1-12-54-5
Carton del Jugador 2: 3-6-12-2
Ingrese numero de la tómbola: 1
Ingrese numero de la tómbola: 6
Ingrese numero de la tómbola: 4
Ingrese numero de la tómbola: 3
Ingrese numero de la tómbola: 54
Ingrese numero de la tómbola: 8
Ingrese numero de la tómbola: 5
Ingrese numero de la tómbola: 12
El ganador es el Jugador 1
```

```
Carton del Jugador 1: 1-12-54-5
Carton del Jugador 2: 3-6-12-2
Ingrese numero de la tómbola: 1
Ingrese numero de la tómbola: 6
Ingrese numero de la tómbola: 4
Ingrese numero de la tómbola: 3
Ingrese numero de la tómbola: 54
Ingrese numero de la tómbola: 2
Ingrese numero de la tómbola: 5
Ingrese numero de la tómbola: 12
Ambos jugadores ganan
```

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

2. El servicio de inteligencia de Pythonia ha estado monitoreando los mensajes de texto que recibe el líder del país de Javapolis, y ha detectado un patrón en éstos, pero no ha podido decifrar qué dicen los mensajes, puesto que requieren de un servicio rápido. Como han estado celebrando sus fiestas patrias, nadie ha tenido tiempo de implementar una solución, por lo que han acudido a ud para que los apoye en esta importante misión.

Ellos detectaron que el líder recibe muchas palabras, pero sólo las que contienen una **clave** en alguna parte, son consideradas en el mensaje. Además, notaron que cuando quieren dar por finalizado un mensaje, usan la palabra **out**. Notar que en Javapolis, sólo trabajan con letras minúsculas.

El servicio de inteligencia logró crear la función `get_pos(palabra, clave)`, la cual devuelve la posición en la cual comienza la clave dentro de la palabra y en caso contrario retorna `-1`. Notar que las posiciones comienzan desde 0. A continuación, se presentan ejemplos de cómo funciona `get_pos`.

```
>>> get_pos('foniwida', 'iwi')
3
>>> get_pos('primavera', 'iwi')
-1
```

En base a lo descrito anteriormente se le solicita a usted lo siguiente:

1. Escriba la función `get_palabra(palabra, clave)` la cual recibe 2 textos. La función debe retornar la palabra descryptada, es decir, sin la palabra clave en su interior. Si no existe la palabra clave en su interior debe retornar `-1`. Asuma que la palabra contiene **a lo más una vez** la clave.

```
>>> get_palabra('foniwida', 'iwi')
fonda
>>> get_palabra('iwien', 'iwi')
en
>>> get_palabra('fiesta', 'iwi')
-1
```

2. Desarrolle un programa que solicite el ingreso de la clave que se usó para encriptar las palabras y luego solicite el ingreso de palabras hasta que se ingrese el texto **out**. En dicho momento, dejará de pedir palabras y desplegará por pantalla el mensaje oculto, desifrado.

```
Ingrese clave: iwi
Ingrese palabra: i wana be
Ingrese palabra: feli iwices
Ingrese palabra: programar
Ingrese palabra: public
Ingrese palabra: stdout
Ingrese palabra: iwilos
Ingrese palabra: class
Ingrese palabra: primavera
Ingrese palabra: cuatroiwi
Ingrese palabra: cueca
Ingrese palabra: i win
Ingrese palabra: out
El mensaje oculto es: felices los cuatro
```

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

3. La comisión organizadora de la semana mechona creó un concurso on line, en el cual sus 3 alianzas deben participar. El concurso consta de una serie de turnos, en los cuales cada alianza debe ingresar una palabra. La palabra con mayor cantidad de vocales gana. Si dos o más alianzas igualan en el máximo de vocales, nadie gana esa ronda. El concurso termina cuando **una** alianza logre ganar x juegos (meta), donde x debe definirse al inicio del juego.

En base a lo anterior, la comisión solicita a usted implementar lo siguiente:

1. Escriba la función `ganador(c1, c2, c3, meta)`, la cual recibe 4 parámetros, la cantidad de juegos ganados por la alianza 1, 2 y 3 y la meta a lograr en el juego. La función debe retornar el número de la alianza ganadora. En caso de no existir ganador, debe retornar el valor entero 0 (cero).

Nota: Tener en cuenta que no pueden existir empates en las alianzas al momento de alcanzar la meta.

```
>>> ganador(2, 5, 3, 5)
2
```

```
>>> ganador(1, 3, 3, 5)
0
```

2. Escriba la función `contar(palabra)` que reciba un string `palabra`. La función debe retornar la cantidad de vocales existentes en la palabra recibida.

```
>>> contar('paralelepipedo')
7
```

```
>>> contar('str')
0
```

3. Desarrolle un programa que solicite la meta del Juego, y luego solicite las palabras de cada alianza por turno, hasta que exista una alianza que logre la meta. **Asuma que las palabras pueden ser ingresadas en mayúsculas o minúsculas.** A continuación se presenta un ejemplo de cómo debería lucir el programa.

```
Ingrese meta del juego: 3
alianza 1: Paralelepipedo
alianza 2: reunA
alianza 3: salIda
alianza 1: Tartamudo
alianza 2: Mauricio
alianza 3: semana
alianza 1: mUrciElAgO
alianza 2: rescate
alianza 3: Salvavida
alianza 1: temperatura
alianza 2: Sala
alianza 3: Certamen
La alianza ganadora es 1
```