



## 501125 Introducción a la Programación Laboratorio de Programación usando Strings

### EQUIPO PROGRAMACIÓN

29 de mayo de 2023

- 1.- Un palíndromo es una palabra o frase que se puede leer igual de derecha a izquierda o de izquierda a derecha, por ejemplo, la frase “amor a roma” es palíndromo. Escriba una función que determine si una palabra o frase ingresada es palíndromo.

**Entradas:** La única entrada a la función es un string conteniendo una palabra o frase.

**Salidas:** La única salida de la función es **True** si el string ingresado contiene una palabra o frase palíndromo, de lo contrario la función debe retornar **False**.

- 2.- Un capicúa es un número entero que se puede leer igual de derecha a izquierda que de izquierda a derecha, por ejemplo, 1234321 es capicúa. Escriba una función que determine si un número entero es capicúa. Utilice la función para detectar palíndromos del ejercicio anterior.

**Entradas:** La única entrada a la función es un número entero positivo.

**Salidas:** La única salida de la función es **True** si el número ingresado es capicúa, de lo contrario la función debe retornar **False**.

- 3.- Cuando un programa Python requiere que se le ingrese un número entero positivo, muchas veces al poner en su entrada:

```
n = int(input('ingrese un número entero:'))
```

el programa falla cuando por error el usuario digita algún símbolo distinto de un dígito. Por tanto, se pide escriba una función que lea un string y que se asegure que todos los símbolos ingresados son dígitos.

**Entradas:** La única entrada a la función es una string de símbolos posiblemente lleno con dígitos.

**Salidas :** La salida de la función es **True** si la cadena de entrada esta compuesta sólo de dígitos y **False** de lo contrario.

**Observación :** En el programa principal, si la cadena de entrada esta compuesta por algún símbolo que no es un dígito se debe marcar el error y reingresar la cadena.

- 4.- Modifique el programa anterior para que se pueda ingresar cualquier número entero (positivos, cero y/o negativos).
- 5.- Se dice que existen personas con una capacidad excepcional que les permite leer una texto aún cuando se le haya quitado intencionalmente sus consonantes. Por ejemplo:

A \*o\*\*i\*ua\*ió\* U\*. \*o\*\*á \*ee\* \*a \*i\*ui\*\*\*e \*\*a\*e \*i\* \*o\*\*o\*a\*\*\*e\*.

Para probarlo, se le pide que escriba una función en Python que reciba una frase y genere la frase en que las consonantes han sido reemplazadas por un símbolo ‘\*’.

---

**Entradas:** La única entrada a la función es una string de símbolos con una frase compuesta de sólo letras y espacios.

**Salidas :** La salida de la función es un string igual al de la entrada pero sin las consonantes.

6.- Los problemas de secuencias numéricas son clásicos en las matemáticas recreativas. La secuencia de números enteros “look and say” fue introducida y analizada por el matemático John Conway. Esta secuencia tiene la forma: 1, 11, 21, 1211, 111221, 312211, .... Su construcción es sencilla, cada término describe al anterior:

- 1 contiene “un uno”, o sea, 11;
- 11 contiene “dos unos”, o sea, 21;
- 21 contiene “un dos y un uno”, o sea, 1211;
- 1211 contiene “un uno, un dos y dos unos”, o sea, 111221;
- 111221 contiene “tres unos, dos doses y un uno”, o sea, 312211; y así se puede continuar sucesivamente

En esta secuencia solo se emplean los números 1, 2, y 3. Otra curiosidad de esta secuencia observada por Conway es que muestra similitud formal con el comportamiento de los elementos químicos. También es curioso que el tamaño de cada secuencia es 1,303577...(constante de Conway) veces mayor que la anterior.

Pero, ¿qué ocurre si en lugar de iniciar con un 1 se inicia con otros valores?

Escriba un programa en Python que dado un valor  $n$  ( $n > 0$ ), determine  $n$  términos de la secuencia “look and say” iniciándose en un valor inicial  $v$  ( $v > 0$ ).