

## Práctica 2

Al momento de entregar su solucionario, deben colocar todas sus soluciones en un único archivo .py

### Estructuras Iterativas:

#### **Problema 1:**

Escribe un programa en Python para encontrar los números que son divisibles por 7 y múltiplos de 5, en el rango de 1500 y 2700 (ambos incluidos).

#### **Problema 2:**

Escriba un programa en Python para construir el siguiente patrón.

```
*
* *
* * *
* * * *
* * * * *
* * * *
* * *
* *
*
```

#### **Problema 3:**

Por medio de un bucle While genere un código que permita el ingreso de números por teclado. El ingreso de los números debe ser permitido hasta que el usuario decida ya no ingresar nuevos números.

Con dichos números, su programa debe evaluar cada uno de estos números e indicar la cantidad de números pares e impares.

#### **Ejemplo:**

```
"Desea ingresar un número?": SI
"Ingrese el número:" 5
"¿Desea ingresar un número?": SI
"Ingrese el número: " 7
.....
"Desea ingresar un número?": NO
```

Números ingresados: [ 5,7, 6, 7, 8, 9, 1, 2, 3, 4]

Cantidad de números pares: 5

Cantidad de números impares: 4

#### **Nota:**

- Quizás a manera de ayuda el uso de una lista le sea de utilidad.
- El empleo de break sobre condiciones while también le serán de utilidad.

#### **Problema 4:**

Imaginemos que lo han contratado para un colegio donde se desea realizar un sistema por el cual se pueda generar un listado de “n” alumnos y 3 calificaciones que corresponden a alguna de sus materias.

Puede usar el siguiente esquema a manera de ejemplo

```
{  
    Alumno: Juan,  
    Notas: [10, 12, 15]  
}
```

Una vez completado el ingreso de los datos, su programa debe mostrar en pantalla el listado completo de los alumnos.

#### **Nota:**

El uso de listas con diccionarios le será de utilidad.

### **Funciones:**

#### **Problema 5:**

Genere una función que tenga como parámetros el ingreso de un número entero y un dígito. Verifique la cantidad de veces que se usa dicho número en el dígito solicitado.

#### **Ejemplo:**

**Número ingresado:** 15789000 y **Dígito:** 0

**Cantidad de veces 0 en el número:** 4

**Nota:** Para resolver este problema, algunos tipos de datos dentro de python contemplan un método count.

#### **Problema 6:**

Escriba un programa en Python para obtener la serie de Fibonacci entre 0 y 50.

Nota: La secuencia de Fibonacci es la serie de números:

0, 1, 1, 2, 3, 5, 8, 13, 21, .. ...

Cada número siguiente se obtiene sumando los dos números anteriores.

#### **Problema 7:**

Escriba una función de Python que tome un número como parámetro y verifique que el número sea primo o no.

#### **Problema 8:**

Escribe una función de Python para calcular el factorial de un número (un entero no negativo). La función acepta el número como argumento.

## Métodos de Cadenas:

### Problema 9:

Al enviar mensajes de texto o twittear, no es raro acortar las palabras para ahorrar tiempo o espacio, por ejemplo, omitiendo las vocales.

Implemente un programa que solicite al usuario una cadena de texto y luego retorne ese mismo texto pero con todas las vocales (A, E, I, O y U) omitidas, ya sea que se ingresen en mayúsculas o minúsculas.

Ejemplo:

- **Input:** Twitter                      **Output:** Twttr
- **Input:** What's your name?   **Output:** Wht's yr nm?

### Problema 10:

En los Estados Unidos, las fechas suelen tener el siguiente formato: mes-día-año (MM/DD/AAAA). Las fechas en ese formato no se pueden ordenar fácilmente porque el año de la fecha es el último en lugar del primero. Intente ordenar, por ejemplo, 2/2/1800, 3/3/1900 y 1/1/2000 cronológicamente en cualquier programa (por ejemplo, una hoja de cálculo). Las fechas en ese formato también son ambiguas. ¡Una fecha como el 8 de septiembre de 1636, podría interpretarse como el 9 de agosto de 1636!

**Implementar un programa que pida al usuario una fecha, en orden mes-día-año, con formato como 8/9/1636 o Septiembre 8, 1636,** donde el mes en este último podría ser cualquiera de los valores en la lista abajo:

```
[  
    "Enero",  
    "Febrero",  
    "Marzo",  
    "Abril",  
    "Mayo",  
    "Junio",  
    "Julio",  
    "Agosto",  
    "Septiembre",  
    "Octubre",  
    "Noviembre",  
    "Diciembre"  
]
```

**Luego, genere esa misma fecha en formato AAAA-MM-DD.**

**Ejemplos:**

- **Input:** 9/8/1636 | **Output:** 1636-09-08
- **Input:** Septiembre 8, 1636 | **Output:** 1636-09-08
- **Input:** 1/1/1970 | **Output:** 1970-01-01

**Nota:**

- Los métodos de listas y string le resultarán de gran utilidad.
- Nota que si empleamos `print(f"{n:02}")`, esta completará con 0 valos a la izquierda de un número