

ELEMENTOS DE COMPUTACIÓN – PRUEBA CORTA 2 – ASINCRÓNICA

INSTRUCCIONES:

- Desarrollar en Python 3 según material estudiado a la fecha: tipos de datos numéricos y booleanos, estructuras de secuencia, condicional y repetición. Los strings se pueden usar para efectos de mensajes de interfaz con usuario. No se permite importar módulos.
- Ponga todas las funciones en un solo programa fuente con el nombre **prueba_corta2_su_nombre.py**. Cada ejercicio debe aparecer con su número en el orden en que está en la evaluación y con el nombre indicado. Puede usar funciones adicionales.
- Subir solución al tecDigital EVALUACIONES / PRUEBA CORTA 2. PARA TODO TRABAJO QUE ENVÍE POR MEDIOS DIGITALES DEBE ASEGURARSE QUE ENVIÓ EL TRABAJO CORRESPONDIENTE AL LUGAR INDICADO.
- Requisito para revisar la evaluación: las funciones deben tener documentación interna (COMENTARIOS EN EL PROGRAMA FUENTE): al menos lo que hace la función, entradas, salidas.
- El estudiante debe ser el autor de la evaluación enviada. El plagio (copiar en lo sustancial obras ajenas dándolas como propias -RAE, 2021-) en cualquier evaluación conlleva la aplicación del reglamento institucional respectivo, artículo 75.
- Fecha de entrega: 22 de abril 2022 11pm.
- Se coordinará día y hora para revisar la evaluación junto con el estudiante el cual mostrará el dominio de la solución implementada desde el punto de vista técnico (uso de conceptos de programación y del lenguaje) así como de la funcionalidad (lo que hace la solución). La revisión puede constar de las siguientes actividades:
 - Revisar esta solución particular
 - Revisar conceptos incluidos en la evaluación
 - Aplicar otras actividades con una complejidad igual o menor a la evaluación.

RECOMENDACIÓN. Siga la metodología de solución de problemas: entender el problema, diseñar un algoritmo, codificar y probar programa. En la etapa de diseño del algoritmo haga un esquema para determinar el comportamiento o patrón del algoritmo para luego proceder con su desarrollo.

NO SE VAYA DIRECTAMENTE A PROGRAMAR EN LA COMPUTADORA, PRIMERO PLANIFIQUE LA SOLUCIÓN HACIENDO ESTE ESQUEMA SUGERIDO.



- 1) Desarrolle la función **imprimir_palindromos** que lea un número n (entero ≥ 1) (5P) e imprima los primeros n números naturales palíndromos (40P). Un número es palíndromo si al leerlo de izquierda a derecha es igual al leerlo de derecha a izquierda, ejemplos: 22, 393. No haga validación de las entradas, éstas se darán de acuerdo a la especificación. La impresión debe ser como se muestra en este ejemplo del funcionamiento (5P):

```
>>> imprimir_palindromos()
Cantidad de palíndromos a imprimir: 50
1 .    0
2 .    1
...
49.   393
50 .  404
```

- 2) Dos números naturales m y n son llamados números amistosos (o números amigos), si la suma de todos los divisores de m (excluyendo a m) es igual al número n , y la suma de todos los divisores del número n (excluyendo a n) es igual a m , donde $m \neq n$.

Ejemplo: los números 220 y 284 son números amistosos.

Divisores de 220 sin incluirlo: 1, 2, 4, 5, 10, 11, 20, 22, 44, 55, 110, y la suma de ellos es: $1 + 2 + 4 + 5 + 10 + 11 + 20 + 22 + 44 + 55 + 110 = 284$.

Divisores de 284 sin incluirlo: 1, 2, 4, 71, 142, y la suma de ellos es: $1 + 2 + 4 + 71 + 142 = 220$.

Desarrolle la función **determinar_numeros_amistosos** que reciba dos números naturales diferentes (≥ 2) y retorne el valor booleano True si son números amistosos (30P), de lo contrario retorne False (10P). Valide las restricciones y en caso de errores retorne los mensajes respectivos (10P). Ejemplos del funcionamiento:

```
>>> determinar_numeros_amistosos (220, 284)
True
```

```
>>> determinar_numeros_amistosos (18, 21)
False
```

```
>>> determinar_numeros_amistosos (1210, 1184)
True
```

```
>>> determinar_numeros_amistosos ("1210", 1184)
ERROR: LAS ENTRADAS DEBEN SER NÚMEROS NATURALES  $\geq 2$ 
```

```
>>> determinar_numeros_amistosos (1210, -1184)
ERROR: LAS ENTRADAS DEBEN SER NÚMEROS NATURALES  $\geq 2$ 
```

```
>>> determinar_numeros_amistosos (1210, 1210)
ERROR: LAS ENTRADAS DEBEN SER NÚMEROS DIFERENTES
```