

ASPECTOS GENERALES DE LA EVALUACIÓN:

- Lea atentamente las instrucciones antes de comenzar a desarrollarla.
- La evaluación equivale a la nota de un quiz, es INDIVIDUAL y OBLIGATORIA. La detección de copia o ayuda de un tercero será sancionada con calificación mínima. Adicionalmente se tendrá en consideración la situación ocurrida si es que vuelve a ocurrir algo en el futuro.
- Recuerde seguir la estructura de los programas en el curso:

#CONSTANTES

instrucciones...

#ENTRADAS

instrucciones...

#PROCESAMIENTO

instrucciones...

#SALIDAS

instrucciones...

- La respuesta consistirá en un archivo .py nombrado de la siguiente forma: nombre_apellido.py.
- El archivo deberá ser enviado al correo john.serrano@usach.cl.
- La fecha límite para enviar su respuesta es el **martes 22 de octubre a las 23:59 horas**. Soluciones enviadas con posterioridad a la hora de entrega máxima NO serán revisadas.
- Considere que cualquier supuesto que el o la estudiante haga, debe ser explicitado en los comentarios del Script.
- Considere que la evaluación contempla el código, los comentarios y el seguimiento de las buenas prácticas de programación.
- Para esta evaluación, se prohíbe el uso de cualquier herramienta de programación no vista en clases. De igual forma, se prohíbe el uso de Inteligencia Artificial. Estudiantes que entreguen una solución que utilicen herramientas como Programación Orientada a Objetos, Diccionarios, Tuplas, Funciones, entre otros o que utilicen ChatGPT, Copilot, Bing, entre otros, serán sancionados con la muerte académica. **NO SE ARRIESGUE.**

Ayuda:

- Operadores aritméticos: `+, -, *, /, //, %, **`
- Operadores de comparación: `>, >=, <, <=, ==, !=`
- Operadores lógicos: `and, or, not`
- Funciones: `int(), float(), str(), bool(), print(), input(), len(), range(), round()`
- Métodos de Strings: `upper(), lower(), count(), isdigit(), isalpha(), isupper(), islower()`
- Operador para saber si un String está contenido en otro: `in`
- Estructuras de control: `if, if-else, if-elif-else, while, for-in`

LOS NÚMEROS PALÍNDROMOS

Se dice que un numero natural es un **palíndromo** cuando mantiene su valor luego de ser invertido. En otras palabras, si el número es equivalente cuando se lee de izquierda a derecha o de derecha a izquierda. Por ejemplo:

- 12321 es palíndromo, pues al invertirlo mantiene su valor.
- 73481 no es palíndromo. Al invertirlo su valor es $18437 \neq 73481$

Por otra parte, un numero se llama **palindrómico** cuando es posible generar un numero palíndromo a partir de el, aplicando la siguiente secuencia de operaciones en forma repetitiva:

- Sumar el numero y su palíndromo.
- Si el resultado **no es** un palíndromo, se repite el proceso con el nuevo numero generado.

Ejemplos:

- El numero 13 es palindrómico, ya que $13 + 31 = 44$.
- El numero 95 es palindrómico, ya que $95 + 59 = 154 \rightarrow 154 + 451 = 605 \rightarrow 605 + 506 = 1111$.
- El numero 196 no es palindrómico.

Implemente un programa que lea como entrada un número entero positivo que sabemos que es palindrómico. El programa debe generar e imprimir como salida el número palíndromo al que se llega usando la secuencia de operaciones explicada anteriormente. Además, debe mostrar cada número intermedio generado y la cantidad de pasos efectuados para llegar al resultado final.

El numero ingresado siempre **será un número palindrómico**, por lo que puede suponer que el procedimiento descrito llegara siempre a un palíndromo. No es necesario considerar otros casos.

Ejemplo de varias ejecuciones:

```
Ingrese un numero palindrómico: 13
Generado en 1 pasos
Palíndromo final: 44
```

```
Ingrese un número palindrómico: 95
Intermedio: 154
Intermedio: 605
Generado en 3 pasos
Palíndromo final: 1111
```

```
Ingrese un número palindrómico: 97
Intermedio: 176
Intermedio: 847
Intermedio: 1595
Intermedio: 7546
Intermedio: 14003
Generado en 6 pasos
Palíndromo final: 44044
```