Instrucciones: ***Usted tiene que mostrar todo su trabajo de forma clara y ordenada para obtener todos los puntos***. *Este certamen consta de 2 preguntas, las cuales serán entregadas de una en una. Sus desarrollos y código sebe ser subido a la plataforma Aula en los tiempos indicados. Puntos parciales serán entregados a preguntas incompletas. Respuestas finales sin desarrollo o* ***sin nombre*** *reciben 0 puntos.* Copy-and-Paste *de algoritmos reciben 0 puntos. ¡Éxito!*

Se recuerda que:

- Al finalizar la totalidad de su evaluación, deberá adjuntar la siguiente *Declaración de Trabajo Individual* **escrita a mano**.

> **Declaración de Trabajo Individual**: *Juro o prometo que la totalidad del trabajo que he entregado en esta evaluación corresponde a mi trabajo individual, y es el fruto de mi estudio y esfuerzo. Además declaro que no he recibido ayuda externa ni he compartido de forma alguna mi trabajo o desarrollos.*
>
> Nombre, Rol, Firma y Fecha: _____

- Si respondieron las 3 preguntas, se debe indicar en conjunto con la *Declaración de Trabajo Individual* qué preguntas deben revisarse dado que solo se revisarán 2 preguntas. Si no hubiera indicación, se le revisarán las últimas 2 preguntas entregadas. Esto no podrá ser modificado de forma posterior a la entrega de su "Declaración de Trabajo Individual".
- **Avanzar a la segunda página para ver la pregunta**.

1. Considere la siguiente serie:

$$\sum_{k=1}^{\infty} k^{-2} = \frac{\pi^2}{6} \approx 1.64493406684822643647241516664602518921894990120679843773555822937000\ldots \tag{1}$$

Suponiendo que uno no conoce el valor de la serie y quisiera obtener una estimación de ella. Por ejemplo, uno puede expandir los términos de la serie de la siguiente forma:

$$\sum_{k=1}^{\infty} k^{-2} = 1 + \frac{1}{2^2} + \frac{1}{3^2} + \frac{1}{4^2} + \cdots + \frac{1}{K^2} + \frac{1}{(K+1)^2} + \ldots \tag{2}$$

Notar que podríamos obtener computacionalmente el valor de la serie (2) sumando los términos de izquierda a derecha, i.e. primero se parte con 1, luego se le suma $\frac{1}{2^2}$, y así sucesivamente. A este algoritmo lo denotaremos como Algoritmo 1. La otra forma de obtener una estimación del valor de la serie es realizar la suma de derecha a izquierda, donde claramente no podemos partir desde infinito! Sin embargo podemos truncar la serie hasta el termino K, es decir, hasta el término $\frac{1}{K^2}$, y luego sumar desde ese termino hasta llegar a 1. Aunque claramente esta segunda opción es una aproximación desde el comienzo, evaluaremos su comportamiento. Este será el Algoritmo 2.

IMPORTANTE:

- Por simplicidad, en esta pregunta se trabajará con ***single precision***, es decir, los números de punto flotante utilizarán 32 bits para su almacenamiento. Recordar que en ***single precision*** se utiliza 1 bit para el signo, 23 bits para la mantisa y 8 bits para el exponente.
- En Python 3 cualquier variable se puede almacenar en ***single precision***, para esto se puede utilizar la función de Numpy `np.float32()`.
- Para implementar los algoritmos en ***single precision*** se debe hacer *casting* luego de cada operación, por ejemplo al multiplicar 2 números en ***single precision*** y mantener el resultado en ***single precision*** uno puede hacer lo siguiente `c=np.float32(a*b)`, otro ejemplo sería elevar un número a a la potencia b: `c=np.float32(a**b)`, y así con todas las operaciones.

(a) ✍, **[10 puntos]** En el Algoritmo 1 sabemos que, por como opera la representación de punto flotante, podemos truncar la serie dado que si estamos sumando números positivos la suma será creciente. Esto nos permite concluir que para cierto K, el seguir agregando términos a la suma será en vano, lo único que significará será tiempo extra de computación. Por esto, se le solicita que estime un valor de K donde se asegura que esto ocurrira, i.e. debe obtener un valor de K donde es seguro truncar la suma al estar trabajando en ***single precision***. Usted debe mostrar todo su desarrollo, respuestas sin justificación no reciben puntaje. *Hint: This estimation does not have to be tight, but it must be an upper bound of the tight estimation.*

(b) ⌨, **[15 puntos]** Implemente el Algoritmo 1 utilizando ***single precision*** (i.e. use `np.float32()`). Su implementación debe recibir K como **input**, y el **output** debe ser un NumPy Array donde el primer término es la aproximación obtenida y el segundo término es el logaritmo en base 10 del valor absoluto del error. El error debe obtenerse respecto a la aproximación entregada en (1). El K a utilizar en esta implementación debe ser el obtenido en la pregunta anterior.

(c) ⌨, **[15 puntos]** Implemente el Algoritmo 2 utilizando ***single precision***. Su implementación debe recibir K como **input**, y el **output** debe ser un NumPy Array donde el primer término es la aproximación obtenida y el segundo término es el logaritmo en base 10 del valor absoluto del error. El error debe obtenerse respecto a la aproximación entregada en (1). El K a utilizar en esta implementación debe ser definido por usted tal que el error obtenido en escala logarítmica (es decir la segunda componente de su **output**) sea menor que -6 (i.e. $\log_{10}(\text{error}) < -6$).

(d) ✍, **[10 puntos]** Dado los resultados obtenidos por los 2 algoritmos implementados, ¿Cómo puede explicar lo ocurrido? ¿Qué conclusión/es puede obtener? Asegúrese de responder ambas preguntas.