

Lista de Ejercicios de Programación #02

01 - Calcular estadísticas (PCD08)

Escriba un programa en el lenguaje Python 3.x que lea una cantidad variable de números enteros mayores o iguales a cero y calcule las siguientes estadísticas:

- `n` - la cantidad de valores
- `media` - la suma de todos los números divididos entre `n`
- `min` - el menor valor ingresado
- `max` - el mayor valor ingresado

El programa debe dejar de realizar lecturas cuando el usuario ingrese un valor negativo y no debe considerar este número. Debe imprimir la media con 2 valores decimales únicamente, el resto de valores se deben imprimir como enteros.

En caso no haya datos deberá imprimir "No hay datos"

El nombre del programa debe ser: *.py

Link: <https://grader.labs.org.pe/web/project/981>

Ejemplo:

Entrada	Salida
-1	No hay datos

Entrada	Salida
6 4 4 2 -1	n=4 media=4.00 min=2 max=6

Entrada	Salida
1 2 3 4 5 6 7 8 -5	n=8 media=4.50 min=1 max=8

02 - Lista de números (PCD09)

Escriba un programa en el lenguaje de programación Python 3.x que lea dos números, a y b, y a continuación imprima la lista de números enteros comprendidos entre a y b (incluidos a y b). Puede asumir que siempre $a \leq b$.

El nombre del programa debe ser: *.py

Link: <https://grader.labs.org.pe/web/project/966>

Ejemplo:

Entrada	Salida
3 5	3 4 5

Entrada	Salida
1 1	1

Entrada	Salida
-5 -2	-5 -4 -3

	-2
--	----

03 - Calcule un elemento de la serie (PCD10)

Escriba un programa en el lenguaje Python 3.x que calcule el término i de la serie s . El término $s(i) = s(i-1) + s(i-2)$, es decir, un término es equivalente a la suma de los dos términos anteriores.

i	1	2	3	4	5	6	7
$s(i)$	0	1	1	2	3	5	8

El programa deberá leer un número entero i , y a continuación deberá imprimir el valor de $s(i)$.

El nombre del programa debe ser: *.py

Link: <https://grader.labs.org.pe/web/project/967>

Ejemplo:

Entrada	Salida
3	1

Entrada	Salida
6	5

04 - Juego de Mesa (PCD11)

Dos amigos se juntan regularmente a jugar un juego de mesa y anotan el nombre del ganador. La cantidad de juegos es variable, pues depende del tiempo libre que tengan, así que para poder determinar al ganador del mes es necesario contar las veces que gana cada uno. No se puede empatar la partida, pero sí el resultado del mes.

Escriba un programa en Python 3.x que ayude a estos amigos a determinar quién ganó en el mes.

Deberá leer el nombre del **jugador 1** y luego el del **jugador 2**. A continuación, leerá el número de juegos que se jugaron N ($1 < N < 100$). Luego habrá N líneas con el nombre del jugador que ganó la partida.

El resultado deberá ser el nombre del jugador que ganó el mes y si ambos tuvieron la misma cantidad de juegos ganados, deberá indicar `empate`.

El nombre del programa debe ser: `*.py`

Link: <https://grader.labs.org.pe/web/project/976>

Entrada	Salida
Maria Pedro 4 Maria Maria Pedro Maria	Maria

Entrada	Salida
Pedro Daniel 6 Daniel Daniel Pedro Daniel Pedro Pedro	empate

05 - Uber: Autos más cercanos (PCD12)



Escriba un programa en el lenguaje de programación Python 3.x que lea una distancia máxima en metros, y luego el nombre y la posición relativa de posibles conductores (nombre, x, y) considerando que el usuario que solicita el servicio está en la posición (0,0).

Si la menor distancia euclidiana fuera mayor a la distancia máxima debe indicar **“muy lejos”** de lo contrario debe indicar **el nombre del conductor más cercano**.

La cadena **FIN** indica el final de la lista. La cantidad de conductores listados será como mínimo 1 y máximo 15. **Hint: Considere que ningún conductor de la lista se encuentra a la misma distancia (es decir, solo existe un mínimo).**

Considere la siguiente entrada:

```
500
Pedro
0
```

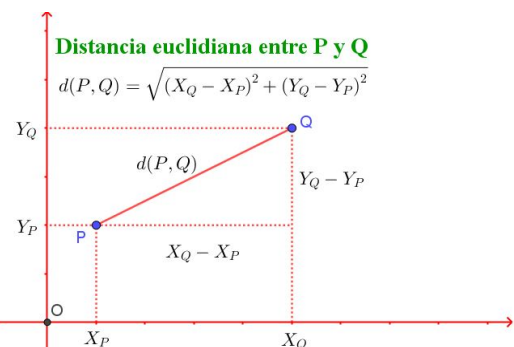
```
600
Juan
0
700
FIN
```

Interpretación

Significa que la distancia máxima es **500m**, habiendo dos posibles conductores: Pedro y Juan, que se encuentran en la posición **(0, 600)** y **(0, 700)** relativas al usuario, respectivamente. Calculando la distancia euclidiana podremos ver que el conductor Pedro se encuentra a **600m** del usuario y el conductor Juan a **700m**, por lo que el programa deberá imprimir “**muy lejos**”

Entrada	Salida
500 Daniel 0 600 FIN	muy lejos
700 Maria 600 0 FIN	Maria
500 Emilio 301 400 Maria 600 0 Pedro 300 400 FIN	Pedro

Hint: puede calcular la distancia euclidiana entre dos puntos usando la siguiente fórmula. Considere el uso de la función `min`¹ y la función `index`²



¹ https://www.tutorialspoint.com/python/list_min.htm

² <https://www.programiz.com/python-programming/methods/>

El nombre del programa debe ser: *.py

Link: <https://grader.labs.org.pe/web/project/977>

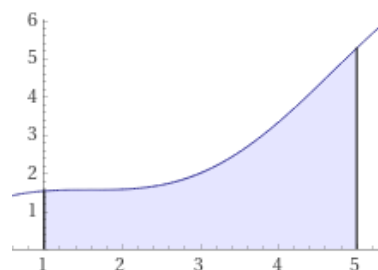
06 - Integral definida (PCD13)

Escriba un programa en el lenguaje Python 3.x que realice el cálculo del área bajo la curva de la función $f(x)$ entre los extremos a y b . Debe leer a y b como dos números enteros.

La función $f(x)$ que usaremos será la mostrada a continuación: $f(x) = \cos(x) + x$
Por ejemplo, si $a=1$ y $b=5$, deberá calcular numéricamente la integral definida

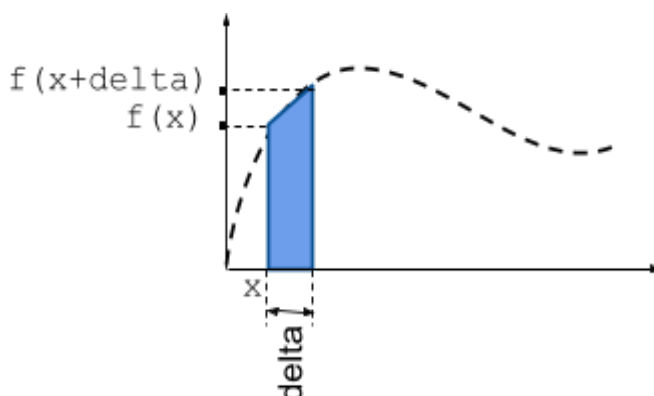
$$\int_1^5 (\cos(x) + x) dx$$

Esto equivale al área sombreada mostrada.



Para realizar esta tarea use el siguiente algoritmo, que aproxima el área bajo la curva como la suma de muchos trapecios pequeños.

1. Iniciar el área total en 0.0
2. Definir $x = a$
3. Definir $\text{delta} = 0.001$
4. Mientras $x < b$ hacer lo siguiente
 - a. Calcular el área³ del trapecio siguiente:



- b. Sumar el área calculada al área total
 - c. Hacer que $x = x + \text{delta}$
5. Imprimir el área total

³ El área del trapecio está dado por $(\text{base_mayor} + \text{base_menor}) * \text{altura} / 2$

Implemente $f(x)$ como una función en Python. Para eso puede usar la librería `math`

Nota: Debe imprimir 2 posiciones decimales únicamente.

El nombre del programa debe ser: *.py

Link: <https://grader.labs.org.pe/web/project/978>

Entrada	Salida
1 5	10.20
8 12	38.49
20 23	62.74

07 - Combinatoria (PCD14)

Escriba una función en el lenguaje Python 3.x que calcule la combinatoria de n en k (diferentes formas de seleccionar grupos de tamaño k de una población de tamaño n) usando la siguiente fórmula:

$$C_{n,k} = \frac{n!}{k!(n-k)!}$$

La función deberá tener la siguiente cabecera

```
combinatoria(n, k)
```

Puede usar la función `math.factorial` o implementar una función auxiliar que calcule el factorial considerando que

$$x! = 1.2.3.4...x$$

La función debe retornar un número entero como resultado. Considere los siguientes ejemplos:

n	k	$C_{n,k}$
5	5	1
10	3	120
9	2	36

Nota: Recuerde que debe retornar el valor y no imprimir el resultado directamente.

El nombre del programa debe ser: *.py

Link: <https://grader.labs.org.pe/web/project/980>

08 - Rectified Linear Unit (PCD15)

Las redes neuronales son uno de los métodos de inteligencia artificial más utilizados hoy en día. Una innovación muy exitosa, y a la vez simple, consiste en aplicar una familia de funciones conocidas como `RELU` (Rectified Linear Unit) reemplazando a funciones de activación más complejas.

Se le pide implementar una función utilizando el lenguaje de programación Python, que teniendo como dato de entrada la salida de una neurona (un único número decimal) devuelva la activación de acuerdo a la función `LEAKY_RELU`, una de las variantes de la activación `RELU`. La función implementada debe tener la siguiente cabecera

```
leaky_relu(x)
```

La función `Leaky RELU` está definida por partes, si el número es positivo devuelve el mismo número y si es negativo se devuelve 0.01 multiplicado por el número decimal recibido como entrada, como puede verse a continuación.

$$f(x) = \begin{cases} x & \text{if } x > 0 \\ 0.01x & \text{otherwise} \end{cases}$$

El nombre del programa debe ser: `funciones.py`

Link: <https://grader.labs.org.pe/web/project/968>

Ejemplo:

Ejemplo de código ejecutado	Salida
<pre>x = 4 lr = leaky_relu(x) print(lr)</pre>	4

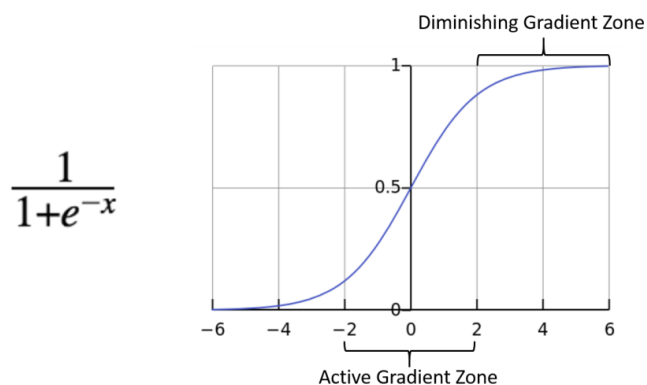
Ejemplo de código ejecutado	Salida
<pre>x = -4 leaky_relu(x) print(lr)</pre>	-0.04

09 - Sigmoid (PCD16)

Escriba un programa en el lenguaje Python 3.x que implemente la función de activación `sigmoid` ampliamente usada en redes neuronales

```
sigmoid(x)
```

Para implementar la función `sigmoid` tendrá que usar la función exponencial, como puede ser visto a continuación. Recuerde que en el paquete `math` se encuentra la función `math.exp`



Fuente: Medium⁴

El nombre del programa debe ser: `funciones.py`

Link: <https://grader.labs.org.pe/web/project/982>

Ejemplo:

Ejemplo de código ejecutado	Salida
<pre>x = 0 print(sigmoid(x))</pre>	0.5

Ejemplo de código ejecutado	Salida
<pre>x = 8 print(sigmoid(x))</pre>	0.9996646498695336

Ejemplo de código ejecutado	Salida
<pre>x = -100 print(sigmoid(x))</pre>	3.7200759760208356e-44

⁴

<https://medium.com/@ionutlang31/how-to-play-around-with-sigmoid-function-to-increase-its-y-max-and-shift-to-the-right-for-positive-ed40daf1fa2>