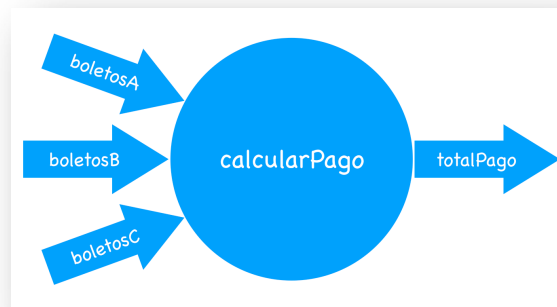


Resuelve los siguientes problemas usando funciones. Antes de empezar a escribir código te sugiero que hagas el análisis de cada problema en papel (no se entrega) y consideres la aproximación Top-Down. Cada problema se resuelve en un archivo de Python independiente con el **nombre indicado en este documento**. El resultado de tus programas **DEBE ser IDÉNTICO** al mostrado, incluyendo espacios, líneas en blanco, número de decimales, etc.

1. Boletos para un concierto, **Boletos.py** (400 puntos)

Hay 3 zonas diferentes en el Auditorio Nacional. Se están vendiendo boletos para el próximo concierto de Maroma 5. Cada boleto para la zona A cuesta \$3250, para la zona B cuesta \$1730 y para la zona C cuesta \$850. Escribe un programa que pregunte al usuario cuántos boletos quiere comprar para cada zona y que imprima el total a pagar.

Escribe la función **calcularPago** que recibe como parámetros el número de boletos que quiere comprar para cada zona y regresa el total a pagar.



```
def calcularPago(boletosA, boletosB, boletosC) :  
    # Calcula y guarda en la variable totalPago el total a pagar  
    # Regresa totalPago
```

En la función **main** pregúntale al usuario la cantidad de boletos que quiere comprar para cada zona y los envías a la función **calcularPago** para que regrese el total a pagar, finalmente imprimes el resultado.

```
def main() :  
    # numeroBoletosA = Leer el número de boletos en zona A  
    # numeroBoletosB = Leer el número de boletos en zona B  
    # numeroBoletosC = Leer el número de boletos en zona C  
    # Llama a la función calcularPago, envía como argumentos los valores leídos. Guarda el resultado.  
    # Imprimir el resultado
```

Finalmente, llama a la función **main** para resolver el problema.

```
main()
```

Aquí hay dos ejemplos de cómo se debe comportar tu programa. Muestra centavos en el resultado. La entrada del usuario se muestra en rojo y cursiva.

Número de boletos en zona A: 2	Número de boletos en zona A: 3
Número de boletos en zona B: 1	Número de boletos en zona B: 0
Número de boletos en zona C: 3	Número de boletos en zona C: 0
El costo total es: \$10780.00	El costo total es: \$9750.00

2. Datos de un trapezio, Trapecio.py (400 puntos)

Escribe un programa que lea la base mayor, la base menor y la altura de un trapezio isósceles y que imprima:

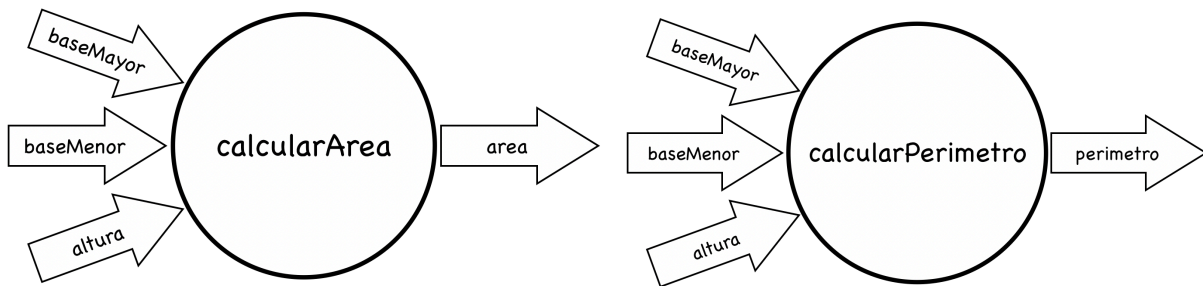
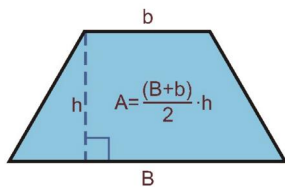
- Área.
- Perímetro.

Debes escribir dos funciones: una para calcular el área y otra para calcular el perímetro.

Escribe la función **main** que lea la base mayor, la base menor y la altura. Llama a las funciones e imprime los resultados con 2 decimales.

Ejemplos:

Escribe la longitud de la base mayor: 2	Escribe la longitud de la base mayor: 10
Escribe la longitud de la base menor: 1	Escribe la longitud de la base menor: 4
Escribe la altura: 1	Escribe la altura: 4
Área: 1.50	Área: 28.00
Perímetro: 5.24	Perímetro: 24.00



3. Cálculo del pago de un trabajador, Pago.py (400 puntos)

El pago semanal de un trabajador se calcula multiplicando las horas normales trabajadas por la cantidad que se paga por hora. Las horas extras se pagan 65% más que las normales.

Escribe un programa que lea las horas normales, las horas extras y el pago por hora de un trabajador. Calcula e imprime los datos de pago semanal. Formatea correctamente las cantidades.

Usa una función para calcular el pago normal y otra función para calcular el pago extra. No olvides la función **main**.

Ejemplos:

Teclea las horas normales trabajadas: 30	Teclea las horas normales trabajadas: 19
Teclea las horas extras trabajadas: 5	Teclea las horas extras trabajadas: 0
Teclea el pago por hora: 50	Teclea el pago por hora: 350
Pago normal: \$1500.00	Pago normal: \$6650.00
Pago extra: \$412.50	Pago extra: \$0.00
-----	-----
Pago total: \$1912.50	Pago total: \$6650.00

Al terminar, sube a github cada uno tus programas y abre un solo Pull request. (No olvides escribir tu nombre y grupo en el mensaje. Si no lo haces, no participas para ganar XP o monedas)

Para la calificación se considerará:

- Comentarios al inicio de cada programa y cada función (para explicar qué hace la función). -50 hp si no están bien.
- Nombres de variables/funciones adecuados de acuerdo con la información que guardan/calculan. -50 hp si no están bien.
- Uso adecuado de funciones para cada problema. -200 hp si no usas funciones.
- Solución correcta del problema (algoritmo). -200 hp si hay errores en la salida.
- Uso de la función **main** como la principal del programa. **Las funciones NO deben leer con *input* y NO deben imprimir con *print*.** Las lecturas y escrituras se hacen en la función **main**. -100 hp si no está bien.