

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. El resultado de tus programas **DEBE ser IDÉNTICO** al mostrado, incluyendo espacios, líneas en blanco, decimales, etc. Crea un proyecto especial para esta misión en Pycharm con Python 3.6/3.7.

### 1. Asientos en un estadio. (300 puntos)

Hay 3 clases de asientos en un estadio de fútbol. La clase A cuesta \$925, la clase B cuesta \$775 y la clase C cuesta \$360. Escribe un programa que pregunte al usuario cuántos boletos quiere comprar para cada tipo de asiento y que imprima el total a pagar.

Usa la función *calcularPago* que recibe como parámetros el número de asientos de cada clase y regresa el total a pagar.

```
def calcularPago(asientosA, asientosB, asientosC) :  
    # Calcula y guarda en la variable totalPago el total a pagar  
    # Regresa totalPago
```

En la función *main* pide al usuario la cantidad de boletos de cada clase de asiento 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 asientos de clase A  
    # numeroBoletosB = Leer el número de asientos de clase B  
    # numeroBoletosC = Leer el número de asientos de clase C  
    # Calcula el resultado con la función calcularPago, envía como argumentos los valores leídos. Guarda el resultado.  
    # Imprimir el resultado
```

Finalmente tu programa 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.

Número de boletos de clase A: <b>2</b>	Número de boletos de clase A: <b>3</b>
Número de boletos de clase B: <b>1</b>	Número de boletos de clase B: <b>0</b>
Número de boletos de clase C: <b>3</b>	Número de boletos de clase C: <b>0</b>
El costo total es: \$3705.00	El costo total es: \$2775.00

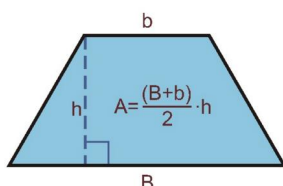
### 2. Datos de un trapecio. (300 puntos)

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

- Área.
- Perímetro.

Debes escribir dos funciones: una para calcular el área y otra para calcular el perímetro. Imprime los resultados con 2 decimales. Ejemplos:

Escribe la longitud de la base mayor: <b>2</b>	Escribe la longitud de la base mayor: <b>5</b>
Escribe la longitud de la base menor: <b>1</b>	Escribe la longitud de la base menor: <b>2</b>
Escribe la altura: <b>1</b>	Escribe la altura: <b>3</b>
Área: 1.50	Área: 10.50
Perímetro: 5.24	Perímetro: 13.71



### 3. Cálculo del pago de un trabajador. (325 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 85% 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 del trabajador incluyendo su pago semanal. Formatea correctamente las cantidades.

Usa una función para calcular el pago normal y otra función para calcular el pago extra.

Ejemplos:

Teclea las horas normales trabajadas: <b>30</b> Teclea las horas extras trabajadas: <b>5</b> Teclea el pago por hora: <b>50</b>  Pago normal: \$1500.00 Pago extra: \$462.50 ----- Pago total: \$1962.50	Teclea las horas normales trabajadas: <b>19</b> Teclea las horas extras trabajadas: <b>0</b> Teclea el pago por hora: <b>350</b>  Pago normal: \$6650.00 Pago extra: \$0.00 ----- Pago total: \$6650.00
-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

### 4. Rendimiento de un auto. (325 puntos)

Debido al aumento de la gasolina, los automovilistas están preocupados por el rendimiento de su auto. Los datos que tienen es el número de kilómetros que recorrieron y la cantidad de litros de gasolina que utilizaron.

Escribe un programa que lea el número de kilómetros recorridos y la cantidad de gasolina utilizada, y que imprima lo siguiente:

- 1) El rendimiento del automóvil en:
  - a. kilómetros/litro.
  - b. millas/galón. (1 milla = **1.6093** kilómetros, 1 litro = **0.264** galones)
- 2) Después, el programa pregunta cuántos kilómetros va a viajar e imprime los litros de gasolina que necesitará.

Debes usar funciones. Utiliza tantas como sean necesarias de acuerdo a tu diseño top-down. Imprime dos decimales.

Ejemplo:

Teclea el número de km recorridos: <b>475</b> Teclea el número de litros de gasolina usados: <b>17</b>  <b>Si recorres 475 kms con 17 litros de gasolina, el rendimiento es:</b> <b>27.94 km/l</b> <b>65.77 mi/gal</b>  ¿Cuántos kilómetros vas a recorrer? <b>250</b>  <b>Para recorrer 250 km. necesitas 8.95 litros de gasolina</b>
-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

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

Para la calificación se considerará:

- Comentarios al inicio de cada programa y cada función (para explicar qué hace la función). -200 hp si no están bien.
- Nombres de variables/funciones adecuados de acuerdo a la información que guardan/calculan. -200 hp si no están bien.
- Uso adecuado de funciones para cada problema. -1000 si no las usan.
- Solución correcta del problema (algoritmo). -900 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. -200 hp si no está bien.