

Programación

# Clase 06

Ejercicios



# Ejercicio 1

La empresa “TicketMusica” necesita un programa para gestionar las ventas de entradas en los diferentes eventos que tiene a cargo. Para ello le ha solicitado a usted que desarrolle el programa, con la condición de que tiene que incluir una función llamada “ventas”. La función debe recibir la cantidad de entradas que se desean comprar y el tipo de entrada. La función debe retornar el monto que se debe cancelar por cada venta.

Los tipos de entradas son las siguientes:

- Cancha: \$20.000
- Platea: \$30.000
- Vip: \$40.000

## Ejercicio 2

Cree un programa en python en el cual se ingresan dos números enteros y una palabra, la cual puede ser **suma** o **multiplicación**.

Una vez ingresados estos valores, debe existir una función llamada **entre** que reciba tanto los números como la palabra y entregue como resultado la suma o la multiplicación de todos los números comprendidos entre ellos (se deben incluir ambos números).

Por ejemplo:

```
print(entre(1, 3, "suma")) # retorna 6
```

```
print(entre(3, 5, "suma")) # retorna 12
```

```
print(entre(1, 2, "multiplicacion")) # retorna 2
```

```
print(entre(5, 8, "multiplicacion")) # retorna 1680
```

## Ejercicio 3

Construye una función llamada **esPrimo** que recibe como parámetro un número, y retorna True o False dependiendo si el número es o no primo.

Por ejemplo:

```
print(esPrimo(7)) # retorna True  
print(esPrimo(6)) # retorna False  
print(esPrimo(7351)) # retorna True
```

Después, usa la función **esPrimo** dentro de un ciclo para escribir por pantalla los números primos entre 3 y 100

## Ejercicio 4 – Parte 1

Construya un programa que informe sobre el número de anillos que aparecen en los dígitos de un número. Para ello deberá crear una función llamada ANILLOS, la cual recibirá un número entero introducido por teclado. Considere lo siguiente:

- Se considera que los dígitos 0, 6 y 9 están formados por un anillo
- Se considera que el dígito 8 está formado por dos anillos.
- El resto de los números no tiene anillos

El programa deberá consultar al usuario si desea evaluar otro número, en el caso de que se ingrese la palabra “No”, el programa terminará.

## Ejercicio 4 – Parte 2

### Ejemplo de ejecución:

Ingrese el numero a evaluar: **10**

Los anillos presentes en el numero son: **1**

¿Desea evaluar otro numero? **SI**

Ingrese el numero a evaluar: **1984082**

Los anillos presentes en el numero son: **6**

¿Desea evaluar otro numero? **SI**

Ingrese el numero a evaluar: **-6342590**

Los anillos presentes en el numero son : **3**

¿Desea evaluar otro numero? **SI**

Ingrese el numero a evaluar: **-24809**

Los anillos presentes en el numero son : **4**

¿Desea evaluar otro numero? **NO**

## Ejercicio 5 – Parte 1

La empresa ABCDEN de Coquimbo está realizando un remate de los últimos productos que le están quedando debido al quiebre de la empresa, por lo cual se requiere construir un programa para obtener información de las ganancias posibles que se pueden obtener.

Para esto, se cuenta con los siguientes archivos:

- artículos.txt: Contiene el nombre del producto y la cantidad disponible (Kgs)
- precios.txt: Contiene el precio de cada producto (\$/Kgs)

Con dicha información usted debe realizar lo siguiente:

Crear una función que permita obtener el total de dinero que se obtendrá del remate. Además, deberá recibir como parámetro el porcentaje de descuento que se le brindará al potencial cliente. El programa debe informar el valor final a cancelar por parte del cliente.

El programa además deberá informar el producto con mayor precio (nombre y valor), para ello deberá crear una función de nombre MAYOR que tenga como funcionalidad obtener el precio mayor.

## Ejercicio 5 – Parte 1

### artículos.txt

Tomate, 20  
Zanahoria, 15  
Cebolla, 10  
Papas, 35  
Lechuga, 10  
Betarraga, 15

### precios.txt

Zanahoria, 1000  
Cebolla, 1200  
Tomate, 800  
Papas, 500  
Lechuga, 300  
Betarraga, 500



## Ejercicio 6 – Parte 1

Imagina lo siguiente: quieres correr una distancia  $d$ . Pero una vez que llegas a la mitad de tu recorrido, te das cuenta que aun te falta la mitad, así que decides incrementar tu velocidad en un 50%. Pero una vez que llegas a la mitad de lo que te faltaba, te das cuenta que aun te falta mucho, así que vuelves a aumentar tu velocidad en un 50%. Y así sucesivamente.

¿Cuánto tiempo te demorará recorrer la distancia?

Todos sabemos que :

$velocidad = distancia / tiempo$

por lo que  $tiempo = distancia / velocidad$

## Ejercicio 6 – Parte 2

Un ejemplo de ejecución (con distancia=100 y velocidad=2):

Vamos a comenzar con d= 100 y v= 2

En recorrer 50.0 me demoro 25.0 con velocidad 2

En recorrer 25.0 me demoro 8.333333333333334 con velocidad 3.0

En recorrer 12.5 me demoro 2.7777777777777777 con velocidad 4.5

En recorrer 6.25 me demoro 0.9259259259259259 con velocidad 6.75

En recorrer 3.125 me demoro 0.30864197530864196 con velocidad 10.125

En recorrer 1.5625 me demoro 0.102880658436214 con velocidad 15.1875

En recorrer 0.78125 me demoro 0.03429355281207133 con velocidad 22.78125

En recorrer 0.390625 me demoro 0.011431184270690443 con velocidad 34.171875

En recorrer 0.1953125 me demoro 0.0038103947568968147 con velocidad 51.2578125

En recorrer 0.09765625 me demoro 0.0012701315856322715 con velocidad 76.88671875

En recorrer 0.048828125 me demoro 0.0004233771952107572 con velocidad 115.330078125

En recorrer 0.0244140625 me demoro 0.00014112573173691905 con velocidad  
172.9951171875

En recorrer 0.01220703125 me demoro 4.704191057897302e-05 con velocidad  
259.49267578125

En recorrer 0.006103515625 me demoro 1.5680636859657674e-05 con velocidad  
389.239013671875

Tiempo total 37.49999215968157

## Ejercicio 6 – Parte 3

Completa las funciones indicadas

```
# Completa la función calcular_tiempo.  
# Debe retornar el tiempo que demora recorrer la distancia "d" a velocidad constante "v"  
#  
def calcular_tiempo(d, v):  
    return 0  
  
# Completa la función dividir_distancia  
# Debe retornar la fracción de la distancia que se ha recorrido, respecto a la distancia  
# (0 sea, hay que dividir distancia en 2)  
#  
def dividir_distancia(distancia):  
    return 0  
  
# Completa la función  
# Recibe la velocidad actual como parámetro, y retorna la nueva velocidad, de acuerdo  
# al enunciado del problema  
#
```

## Ejercicio 6 – Parte 4

Completa las funciones indicadas

```
def aumentar_velocidad(v):  
    return 0  
  
# Realiza el proceso de iterar, tal como dice el enunciado (recorre la mitad de la  
# distancia, acelera).  
# Este proceso no se puede repetir infinitamente: debe ser controlado para que no se  
# convierta en un ciclo infinito.  
# Ten cuidado con eso. Repite el ciclo mientras la distancia remanente que te quede  
# sea > a 0.01.  
# Debe retornar el tiempo total usado en recorrer la distancia.  
#  
def iterar(d, v):  
    tt = 0  
    return tt  
  
def main():  
    d = 100  
    v = 2  
    print("Vamos a comenzar con d=", d, "y v=", v)  
    tiempo = iterar(d, v)  
    print("Tiempo total", tiempo)
```