

Actividades de la Semana 3

Los alumnos del curso de “Programación para la Física y Astronomía” tienen 8 horas consideradas como trabajo personal establecidos en el programa de la asignatura.

Introducción al Problema 1

Para introducir en el interior de una cadena de caracteres algunos caracteres especiales imposibles de representar mediante texto, utilizamos lo que comúnmente se denominan secuencias de escape.

Todas ellas comienzan con una barra invertida (\) seguida de otro carácter. En tiempo de ejecución, las secuencias de escape se sustituyen por los caracteres adecuados. Estas son las secuencias de escape más usadas:

Carácter de escape	Descripción
\'	Comillas simples
\"	Comillas dobles
\t	tabulador. Produce el mismo efecto que pulsar la tecla del Tabulad
\n	nueva línea. Provoca el salto a la línea siguiente.
\r	retorno de carro. Provoca el retroceso al comienzo de la línea.
\\	Barra invertida

Las secuencias de escape \r y \n pueden ser utilizadas en textos estáticos multilíneas, en los textos con Python usaremos \n para una nueva línea. Utilice `print("abcdefg\r123")` para observar cómo funciona \r

Es importante tener presente que los archivos de texto un carácter de fin de línea, que depende del sistema operativo, se muestre con la secuencia de escape adecuada: \r\n en Windows, \n en Linux y \r en Mac.

Problema 01

Escriba un programa que muestre en pantalla un texto que utiliza los caracteres de escape.

Problema 02

Escriba un programa que solicite al usuario dos números y los almacene en dos variables. En otra variable, almacene el resultado de la suma de esos dos números y luego muestre ese resultado en pantalla.

Problema 03

Escriba un programa que solicite al usuario ingresar la cantidad de kilómetros recorridos por una motocicleta y la cantidad de litros de combustible que consumió durante ese recorrido. Mostrar el consumo de combustible por kilómetro.

Problema 04

Escriba un programa que solicite al usuario un número y le reste el 15%, almacenando todo en una única variable. A continuación, mostrar el resultado final en pantalla.

Problema 05

Escriba un programa que solicite al usuario un número y después se imprima la siguiente secuencia, para el ejemplo el número ingresado por el usuario es 8.

4	6	8	10	12
6	8	10	12	14
8	10	12	14	16
12	14	16	18	20
14	16	18	20	22

Introducción al Problema 06

Las cadenas en Python se pueden declarar encerradas entre comillas simples ('), comillas dobles ("), triples comillas simples (') o triples comillas dobles (').

Las comillas simples y las dobles permiten declarar cadenas de **una sola línea**:

<code>cadena1 = 'Hola Mundo A'</code>	<code>cadena2 = "Hola Mundo B"</code>
---------------------------------------	---------------------------------------

Las comillas triples (tanto simples como dobles) permiten declarar cadenas de **varias líneas**:

<code>cadena3 = '''Hola Mundo C'''</code>	<code>cadena4 = """Hola Mundo D"""</code>
---	---

Las comillas triples dobles, además, sirven para escribir comentarios de varias líneas en el código.

```
"""Esto es un comentario de varias  
líneas y el intérprete lo ignorará"""
```

len(cadena): Devuelve el número de caracteres que contiene la cadena.

el operador de concatenación (+) permite concatenar dos cadenas (cad1 + cad2)

el operador de repetición (*) permite repetir una cadena tantas veces como se le indique tras el operador (cadena * nRepeticiones)

<pre>cadena5 = cadena1 * 3 print(cadena5)</pre>	<pre>cadena6 = cadena1 + cadena2 print(cadena6)</pre>
---	---

Problema 06

Escriba un programa que solicite al usuario primero su nombre y después su apellido en dos variables. Luego muestre en la pantalla el número de caracteres del nombre y del apellido, muestre el nombre completo en la pantalla. Presente los datos en la pantalla de forma clara, que cualquiera entienda que representan los datos.

Problema 07

Escriba un programa que solicite al usuario su nombre. Luego muestra en la pantalla la repetición del nombre 5 veces hacia abajo, utilice el * y carácter de escape.

Introducción al Problema 08

Los caracteres de una cadena ocupan posiciones consecutivas en memoria.

Los índices permiten acceder a los caracteres de la cadena (solo lectura) por posición. Las posiciones de la cadena se empiezan a contar desde 0.

0	1	2	3	4	5	6	7	8	9
H	o	l	a		m	u	n	d	o

Se utiliza el **operador corchetes [índice]** para indicar la posición a la que se quiere acceder.

cadena [**start:end:steps**]: Devuelve una nueva cadena con los caracteres de la original desde la posición start incluida hasta la posición end excluida con steps saltos.

<pre>Cadena7='Hola mundo' print(cadena7) print("cadena7[3]\t",cadena7[3]) print("cadena7[7]\t",cadena7[7])</pre>	<pre>print("cadena7[3:]\t",cadena7[3:]) print("cadena7[:3]\t",cadena7[:6]) print("cadena7[3:6]\t",cadena7[3:6]) print("cadena7[::2]\t",cadena7[::2]) print("cadena7[1:6:2]\t",cadena7[1:6:2])</pre>
--	---

En ocasiones es necesario limpiar la terminal que estamos usando para presentar los datos en una pantalla limpia, recordando lo estudiado en los comandos básicos de Linux podemos usar el comando “clear”.

Abrir el terminal de Ubuntu o Msys2 y ejecute el comando “clear”, en Windows abra un terminal con cmd o powershell y escriba el comando “cls”.

Para borrar la pantalla usando la librería os de Python debemos colocar el siguiente código.

#Para Unix/Linux/macOS/BSD import os os.system ("clear")	#Para DOS/Windows import os os.system ("cls")
--	---

Problema 08

Escriba un programa que solicite al usuario su nombre. Luego muestra en la pantalla el nombre escrito según la forma indicada.

- a) de atrás para adelante en forma horizontal.
- b) Hacia abajo, de adelante hacia atrás
- c) Hacia abajo, de atrás hacia adelante.

Debe escribir 3 programas distintos para cada caso. (ver ejemplo)

Ingrese su nombre: Ale	eIA	A L e	e l A
------------------------	-----	-------------	-------------

Problema 09

Reescriba el problema anterior, donde se solicita el nombre del usuario, después aparece un menú que indica al usuario cuál de las operaciones desea realizar con el nombre ingresado. Después de mostrar el nombre con la opción seleccionada solicita el nombre de otro usuario y aparece el menú.

Deberá añadir una opción de salir del menú.

Introducción al Problema 10

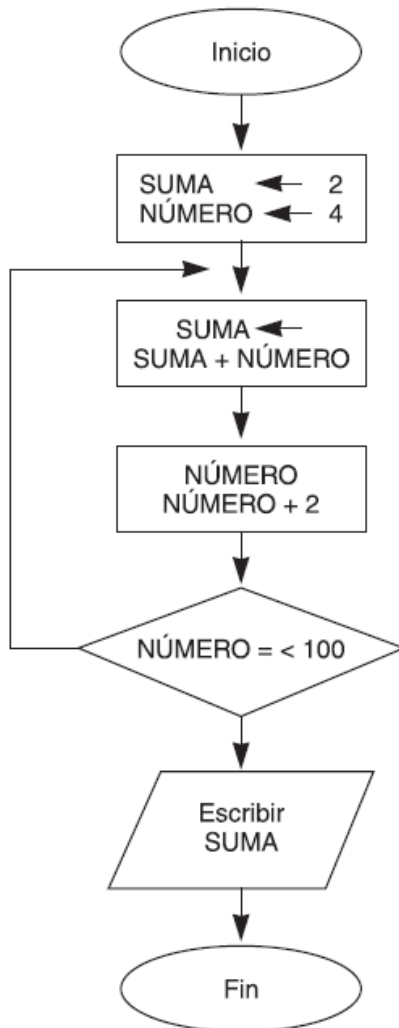
En los siguientes problemas debe implementar el programa en el Python que se encuentra escrito en Pseudocódigo y que tiene el Diagrama de flujo, se recomienda leer el apunte “ESCRITURA DE ALGORITMO”.

Problema 10

Usando el Pseudocódigo reescriba el programa en el lenguaje Python.

El diagrama de flujo es la solución de la suma los numero pares comprendidos entre 2 y 100

Diagrama de flujo



Pseudocódigo

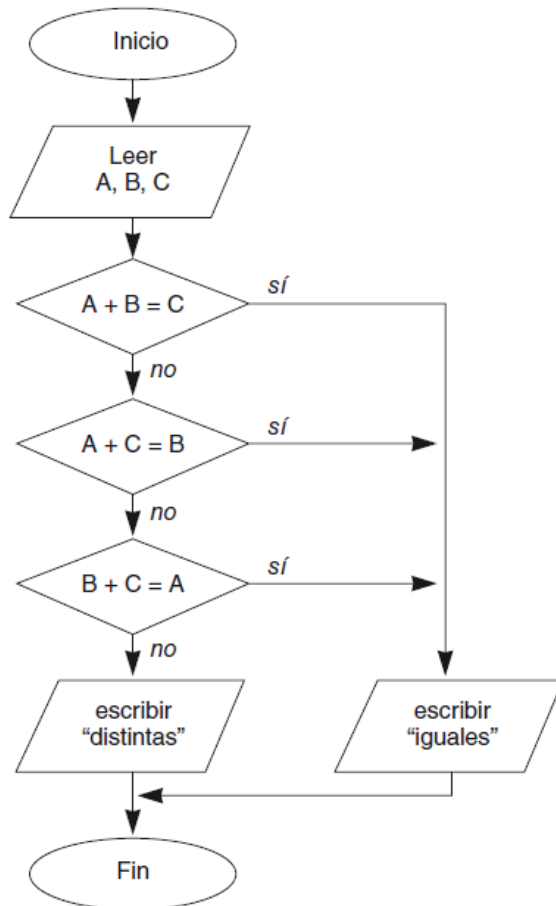
```
entero:    numero, Suma
Suma ← 2
numero ← 4
mientras (numero ≤ 100) hacer
    suma ← suma + numero
    numero ← numero + 2
fin mientras
escribe ('Suma pares entre 2 y 100 =', suma)
```

Problema 11

Usando el Pseudocódigo reescriba el programa en el lenguaje Python.

El diagrama de flujo es la solución de dados tres números, determinar si la suma de cualquier pareja de ellos es igual al tercer número. Si se cumple esta condición, escribir “Iguales” y, en caso contrario, escribir “Distintas”.

Diagrama de flujo



Pseudocódigo

```
entero: a, b, c

inicio
escribir ('test con tres números:')
leer (a, b, c)

si (a + b = c) entonces
    escribir ('Son iguales', a, '+', b, '=', c)
sino si (a + c = b) entonces
    escribir ('Son iguales', a, '+', c, '=', b)
sino si (b + c = a) entonces
    escribir ('Son iguales', b, '+', c, '=', a)
sino
    escribir ('Son distintas')
fin si
fin si
fin si
fin
```

Introducción al Problema 12

Los archivos de texto sin formato a menudo se crean con el formato de archivo de texto más básico, que toma la extensión ".txt". Estos archivos a menudo son creados y editados por el Bloc de notas, el editor de texto que se encuentra en todos los dispositivos Windows, o por otro editor de texto.

Podemos abrir el fichero con la función `open()` pasando como argumento el nombre del fichero que queremos abrir.

Es posible también leer un número de líneas determinado y no todo el fichero de golpe. Para ello hacemos uso de la función `readline()`. Cada vez que se llama a la función, se lee una línea.

El `while` se ejecuta mientras la línea sea distinta de fin de archivo y suma solo los números pares

<pre>fichero = open(datos.txt') print(fichero.readline()) print(fichero.readline())</pre>	<pre>f = open("datos.txt", "r") linea=f.readline() suma_par=0 while linea: if int(linea)%2==0: suma_par+=int(linea) linea=f.readline() print("La suma de los numeros que son pares es",suma_par)</pre>
---	--

Problema 11

Baje el archivo `datos.txt` y realice un programa para cada una de las condiciones establecidas en el procesamiento de datos y genere la mejor salida de los datos procesados en la pantalla, coloque los textos necesarios que identifican lo calculado. El archivo de `datos.txt` contiene números enteros, puede revisar su contenido usando un editor de texto.

- Entregue el resultado de la suma de los números pares y la cantidad de números pares, al mismo tiempo sume los números impares y determine la cantidad de ellos.
- Calcule $\bar{X} \pm \Delta X$ usando los datos, estas fórmulas son conocidas del curso de Modelos Fisicomatemáticos

$\bar{X} = \frac{1}{N} \sum_{i=1}^N X_i$	$\sigma_d = \sqrt{\frac{\sum X_i^2 - \frac{(\sum X_i)^2}{N}}{N-1}}$	$\Delta X = \frac{\sigma_d}{\sqrt{N}}$
--	---	--

- Imprima los datos que son números divisibles por 5 y que en que numero de línea se encuentran en el archivo de datos.
- Identifique los datos que son números divisibles por 4 y súmele 3 unidades, al mismo tiempo identifique los datos que son divisibles por 13 y multiplique el numero por 2. Imprima los números procesados en la pantalla.