

Problema 1: Explique el resultado de cada una de las siguientes expresiones booleanas sabiendo que C tiene valor 41:

- | | |
|---------------------------------|------------------------------------|
| ▪ $C == 40$ | ▪ True and False |
| ▪ $C != 40 \text{ and } C < 41$ | ▪ False or True |
| ▪ $C != 40 \text{ or } C < 41$ | ▪ $\text{False or False or False}$ |
| ▪ $\text{not } C == 40$ | ▪ $\text{True and True and False}$ |
| ▪ $\text{not } C > 40$ | ▪ $\text{False} == 0$ |
| ▪ $C <= 41$ | ▪ $\text{True} == 0$ |
| ▪ not False | ▪ $\text{True} == 1$ |

Problema 2: Utilice un bucle **while** para escribir un programa que imprima en una tabla grados Fahrenheit 0,10,20,...,100 en la primera columna y los correspondientes grados Celcius en la segunda columna. Para este propósito modifique el programa del problema 6 de la guía 1.

Problema 3: Usando un bucle **while**, escriba un programa que genere números impares desde 1 hasta n y los imprima en pantalla. Asegure que si n es un número par, entonces genere a lo sumo $n - 1$ números impares.

Problema 4: Haga un código que compute la suma:

$$s = \sum_{k=1}^M \frac{1}{k}$$

Ayuda: El siguiente código debería hacer esta suma, pero tiene un error. Reescriba este programa correctamente y pruebe variar M . ¿Qué puede decir sobre la convergencia de la suma? ¿Qué ocurre si en vez de sumar $1/k$, suma ahora

```
s = 0; k = 1; M = 100
```

```
while k <= M:
```

```
    s += 1./k
```

```
print s
```

Problema 5: Computando el valor de π . Teniendo en cuenta que:

$$\int_0^x \frac{dt}{\sqrt{1-t^2}} = \arcsin(x); \quad \text{con } -1 \leq x \leq 1,$$

puede mostrarse que la serie de potencias:

$$\arcsin(x) = x + \sum_{n=1}^{\infty} \frac{1 \cdot 3 \cdot 5 \cdots (2n-1)}{2 \cdot 4 \cdot 6 \cdots 2n} \frac{x^{2n+1}}{2n+1},$$

converge en el intervalo $(-1, 1)$. En particular, esta serie vale $\pi/6$ para $x = 1/2$.

Haga un programa que calcule un valor aproximado de π truncando la serie a un número N dado de términos.

Problema 6: Calcule el valor aproximado del número e a partir de la serie de Taylor para e^x :

$$e^x = \sum_{n=0}^{\infty} \frac{x^n}{n!}.$$

Haga el cálculo truncando la serie a 10, 100 y 1000 términos. Compare los resultados con el valor correcto de e .

Problema 7: Haga un programa que, para un dado $x > 1$, permita calcular una aproximación al valor de $\log(x)$ con un error menor o igual a 10^{-3} . A tal fin recuerde la serie de Taylor

$$\log(1+s) = - \sum_{n=1}^{\infty} (-1)^n \frac{s^n}{n}, \quad (\text{logaritmo natural})$$

y el teorema del error para series alternantes truncadas. El programa debe retornar: La cantidad de términos sumados, el valor aproximado de $\log(x)$, una cota para el error de la aproximación obtenida y el valor de $\log(x)$ dado por el módulo `math`. Haga el cálculo para $x = 1.1$, $x = 2.0$ y $x = 11.0$.

Problema 8: Modifique el programa del problema 3 para guardar los números impares en una lista. Para hacer esto, comience el programa definiendo una lista vacía, utilice un bucle `while` y en cada ciclo utilice el método `append` para anexar cada número nuevo a la lista.

Problema 9: Alojar una lista con los números 2,3,5,7,11,13 a una variable llamada `primos`. Escriba los valores de los elementos de la lista usando un bucle `for`.

Problema 10: Suponga que necesita generar coordenadas x uniformemente espaciadas entre 1 y 2 con un espaciado de 0.01. Las coordenadas están dadas por la fórmula $x_i = 1 + ih$, en donde $h = 0.01$ y $0 \leq i \leq 100$. Compute los valores de x_i y almcénelos en una lista (use un bucle `for` y anexe cada nuevo x_i en la lista, la cual debe estar inicialmente vacía).

Problema 11: Listas anidadas. Defina la siguiente lista:

```
q = [['a', 'b', 'c'], ['d','e','f'], ['g','h']]
```

Utilice un doble bucle `for` para poder imprimir en pantalla de a uno cada elemento de la lista.

Problema 12: Explorando que tan chico puede ser un número distinto de cero en una computadora. Escriba el siguiente código:

```
eps = 1.0
while 1.0 != 1.0 + eps:
    print('eps distinguible de cero: %e' % (eps))
    eps = eps/2
```

Explique con palabras que está haciendo el código anterior línea por línea. Examine la salida. Cómo puede ser que $1 \neq 1 + \text{eps}$ no sea cierta? Si alguien le muestra en una sesión interactiva de Ipython que

```
0.5+1.45e-22 = 0.5
```

y le dijera que Python no puede sumar números correctamente, cuál sería su respuesta?

Problema 13: Resuelva el problema 10 usando listas por comprensión.

Problema 14: La relación entre las temperaturas “Celsius”, “Fahrenheit” y “Kelvin” es la siguiente:

$$T_F = \frac{9}{5}T_C + 32, \quad T_K = T_C + 273.15$$

Se desea realizar una lista de listas (listas anidadas) en Python3, donde cada elemento se corresponda con un renglón de una tabla de conversión de temperaturas a tres columnas. La primer columna debe contener las temperaturas Kelvin entre 0K y 500K a incrementos de 20K; la segunda columna las correspondientes temperaturas Celsius y la tercera las correspondientes temperaturas Fahrenheit.

Escriba un script de Python3 que use lisas por comprensión, loops `for` y listas anidadas, para generar la lista de listas (llamada “Temps”). Imprima “Temps” en pantalla usando `pprint.pformat` y luego imprima la tabla de conversión de temperaturas en formato claro y prilijo con valores que muestren dos decimales.

Problema 15: El siguiente texto es impreso en pantalla por el interprete ipython3

```
In [21]: print(NUMS)
[[0, 0, 0], [1, 1, 1], [2, 4, 8], [3, 9, 27], [4, 16, 64], [5, 25, 125], [6, 36, 216], [7, 49, 343], [8, 64, 512], [9, 81, 729], [10, 100, 1000]]
```

Indique qué comandos simples producirían las siguientes salidas (sublistas de NUMS), note que puede haber más de una respuesta válida en cada caso.

- a) Out[22]: [[0, 0, 0], [1, 1, 1], [2, 4, 8]]
- b) Out[23]: [[7, 49, 343], [8, 64, 512], [9, 81, 729], [10, 100, 1000]]
- c) Out[24]: [[2, 4, 8], [3, 9, 27]]
- d) Out[25]: [36, 216]
- e) Out[26]: [8, 64]
- f) Out[27]: [9, 81, 729]
- g) Out[28]: 512

Problema 16: Escriba un script de Python3 que genere el objeto NUMS del ejercicio anterior, lo imprima en pantalla y luego imprima una tabla, a solamente dos columnas, que imprima los cuadrados y cubos extraídos de NUMS.

Problema 17: Escriba un script de Python3 que sirva para generar una lista de listas con los apellidos y nombres de un grupo de personas. El script debe preguntar al operador primero de cuántas personas se trata. Luego debe preguntar al usuario el apellido y nombre (en ese orden) de cada persona y guardar listas de la forma: [Apellido, Nombre] en una lista mayor. Al final debe imprimir la lista de listas en pantalla.

Problema 18: Sea L un lista en Python3. Explique la diferencia entre los dos comandos:

- a) `C = L`
- b) `C = L[:]`