

<https://github.com/TheInventorist/Material-Programacion>

Guia de ejercicios en Python

Ciclos

Javier Aguilera
Vincent Depassier
Roberto León

June 1, 2020

1. Escriba un programa que genere un numero al azar desde 0 a 99 y pregunte al usuario constantemente hasta que lo adivine, cada vez que el usuario pregunte por el numero, el programa debe decirle si el numero es mayor o menor al valor ingresado, también debe señalar cuantos intentos realizo.

```
Ingrese numero: 5
El numero es mayor
Ingrese numero: 82
El numero es menor
Ingrese numero: 75
Correcto , el numero es: 75, lo logro en 3 intentos
```

2. Programe un script que calcule el factorial de un número n . Recuerde que la fórmula para calcular el factorial de un número n es:

$$n! = 1 \times 2 \times 3 \times \cdots \times (n - 1) \times n$$

3. Un grupo de tres niños se sienta en un círculo. Inicialmente, cada niño tiene un número par de chocolates. Cuando el profesor hace sonar una campana, cada niño le da la mitad de sus chocolates a su vecino de la izquierda (en forma simultánea). Después de cada intercambio de chocolates, el profesor le entrega un chocolate extra a cada niño que se quedó con un número impar de chocolates. El juego termina cuando todos los niños tienen la misma cantidad de chocolates.

Desarrolle un programa que lea cuántos chocolates tiene cada niño inicialmente y calcule (y escriba) cuántos chocolates tienen al final, y cuántas veces se debió hacer sonar la campana.

Por ejemplo, si Niño 1 comienza con 2 chocolates, Niño 2 con 4 y Niño 3 con 6, entonces cada niño termina con 6 chocolates y el profesor tocó la campana 3 veces.

4. Desarrolle un programa que calcule y escriba la suma de la serie de Fibonacci hasta el término n (a leer). La serie de Fibonacci viene dada por:

$$\begin{aligned} F_0 &= 0 \\ F_1 &= 1 \\ F_k &= F_{k-1} + F_{k-2} \quad \text{cuando } k \geq 2 \end{aligned}$$

5. Desarrolle un programa que determine si un triángulo es **rectángulo**, **obtusángulo** o **acutángulo**. Recuerde que un triángulo es rectángulo si uno de sus ángulos es recto 90° , es obtusángulo, si uno de sus ángulos es $> 90^\circ$ y acutángulo si todos sus ángulos son $< 90^\circ$, además su algoritmo debe pedir nuevamente los ángulos si no son válidos para un triángulo.

6. En las competencias de clavados, cada salto es evaluado por un panel de siete jueces. Cada juez entrega una puntuación en una escala de 1 a 10, con incrementos de 0.5 . La puntuación más alta y la más baja son eliminadas. La suma de los cinco puntajes restantes es multiplicada por 0.6 , y el resultado es multiplicado por el grado de dificultad del salto. El valor obtenido es el puntaje total del salto. Desarrolle un programa que lea el grado de dificultad del salto y los 7 puntajes de los jueces, para luego mostrar por pantalla el puntaje total del salto. A continuación se muestra un ejemplo:

Grado de dificultad:	3.0
Juez 1:	5.0
Juez 2:	5.5
Juez 3:	4.0
Juez 4:	5.0
Juez 5:	4.5
Juez 6:	5.5
Juez 7:	5.0
El puntaje total es 45.0	

7. Desarrolle un script que **calcule** y escriba todas las combinaciones que den por resultado un número n natural (≥ 6) al sumar 3 números naturales **distintos**.

Ejemplo con $n = 10$: $1 + 2 + 7$, $1 + 3 + 6$, $1 + 4 + 5$, $2 + 3 + 5$.

8. El cifrado César consiste en aumentar cada dígito de un número en un valor determinado (llave), por ejemplo, si el número fuese 16753 y se codificara mediante cifrado César con llave = 1, el número quedaría 27864, como se aprecia cada dígito fue cambiado por el dígito siguiente, ya que el valor de la llave es 1. Cabe señalar que el conjunto se considera circular, es decir, si por ejemplo se quisiera codificar el número 38492 mediante cifrado César con llave = 4, el resultado sería 72836.

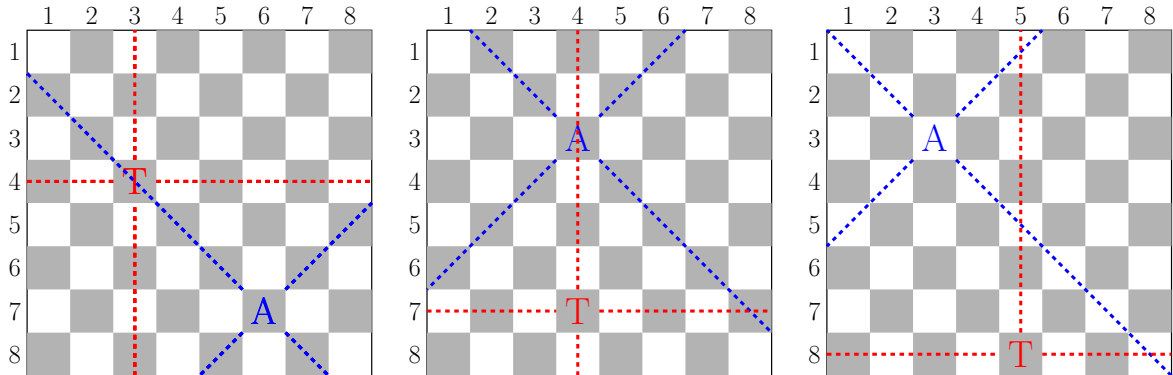
Desarrolle un programa que calcule el cifrado César de un número natural de 10 dígitos. Su algoritmo debe leer tanto el número natural como la llave del cifrado César.

9. El juego del cuarteto es un juego sencillo que tiene 8 intentos para ganar. Cada intento consiste en ingresar cuatro números entre el rango $N > 10$ y $N < 100$.

El jugador gana cuando se logran 3 intentos exitosos. Un intento es exitoso cuando se han ingresado 2 números pares consecutivos y luego 2 números impares consecutivos (o viceversa).

Desarrolle un programa que permita jugar el cuarteto, asuma que todos los números ingresados son válidos, además, su algoritmo debe mostrar un mensaje “Usted ha ganado” cuando el jugador gana el cuarteto, de lo contrario debe mostrar “Usted ha perdido”.

10. Un tablero de ajedrez es una grilla de ocho filas y ocho columnas, numeradas de 1 a 8. Dos de las piezas del juego de ajedrez son el alfil y la torre. El alfil se desplaza en diagonal, mientras que la torre se desplaza horizontal o verticalmente. Una pieza puede ser capturada por otra si está en una casilla a la cual la otra puede desplazarse:



Construya una función que reciba como entrada las posiciones en el tablero de un alfil y de una torre, e indique cuál pieza captura a otra:

```
Fila alfil: 7
Columna alfil: 6
Fila torre: 4
Columna torre: 3
Alfil captura
```

```
Fila alfil: 3
Columna alfil: 4
Fila torre: 7
Columna torre: 4
Torre captura
```

```
Fila alfil: 3
Columna alfil: 3
Fila torre: 8
Columna torre: 5
Ninguna captura
```

Suponga que todos los datos ingresados son válidos. Su función debe funcionar para tableros de 100×100 .

11. Un viajero desea saber cuánto tiempo tomó un viaje que realizó. Él tiene la duración en minutos de cada uno de los tramos del viaje.

Construya un programa que permita ingresar los tiempos de viaje de los tramos y entregue como resultado el tiempo total de viaje en formato **horas:minutos**.

El programa deja de pedir tiempos de viaje cuando se ingresa un 0.

```
Duracion tramo: 51
Duracion tramo: 17
Duracion tramo: 0
Tiempo total de viaje 1:08 horas
```

```
Duracion tramo: 15
Duracion tramo: 30
Duracion tramo: 87
Duracion tramo: 0
Tiempo total de viaje 2:12 horas
```

12. Un número en terciario, es una secuencia de dígitos que pueden ser 0, 1 o 2. Un número terciario puede ser convertido al número entero que corresponde de la siguiente manera:

$$\begin{array}{rcl} 2012 & \longrightarrow & 2 \times 3^3 + 0 \times 3^2 + 1 \times 3^1 + 2 \times 3^0 = 59 \\ 011022 & \longrightarrow & 0 \times 3^5 + 1 \times 3^4 + 1 \times 3^3 + 0 \times 3^2 + 2 \times 3^1 + 2 \times 3^0 = 116 \end{array}$$

Construya un programa que recibe como entrada un número terciario e imprima por pantalla el número entero que le corresponde.

```
Ingrese numero terciario: 2012
El numero terciario 2012 corresponde al numero 59.

Ingrese numero terciario: 011022
El numero terciario 011022 corresponde al numero 116.
```

13. **Asteriscos.**

- (a) Escriba un programa que pida al usuario ingresar la altura y el ancho de un rectángulo y lo dibuje utilizando asteriscos:

```
Altura: 3
Ancho: 5
*****
*****
*****
```

- (b) Escriba un programa que dibuje el triángulo del tamaño indicado por el usuario de acuerdo al ejemplo:

```
Altura: 7
*
**
***
****
*****
*****
*****
```

- (c) Escriba un programa que dibuje el hexágono del tamaño indicado por el usuario de acuerdo al ejemplo:

```
Lado: 4
****
*****
*****
*****
*****
*****
****
```

14. Construya un programa para estimar el valor de π usando la siguiente suma infinita:

$$\pi = 4 \left(1 - \frac{1}{3} + \frac{1}{5} - \frac{1}{7} + \cdots \right)$$

La entrada del programa debe ser un número entero n que indique cuántos términos de la suma se utilizará.

```
n: 5
3.33968253968
```

```
n: 500
3.13959265559
```

```
n: 1000
3.14059265384
```

15. El número de Euler, $e \approx 2.718281828459045$, puede ser representado como la siguiente suma infinita:

$$e = \frac{1}{0!} + \frac{1}{1!} + \frac{1}{2!} + \frac{1}{3!} + \frac{1}{4!} + \cdots$$

Desarrolle un programa que entregue un valor aproximado de e , calculando esta suma hasta que la diferencia entre dos sumandos consecutivos sea menor que la tolerancia ingresada por el usuario.

Recuerde que el factorial $n!$ viene dado por

$$n! = \prod_{i=1}^n i$$

```
Ingrese tolerancia: 0.01
2.71666666667
```

```
Ingrese tolerancia: 0.00001
2.71828152557
```

16. La secuencia de Collatz de un número entero se construye de la siguiente forma:

- si el número es par, se divide por dos;
- si es impar, se le multiplica tres y se le suma uno;
- la sucesión termina al llegar a uno.

La **conjetura de Collatz** afirma que, al partir desde cualquier número, la secuencia siempre llegará a 1. A pesar de ser una afirmación a simple vista muy simple, no se ha podido demostrar si es cierta o no.

Usando computadores, se ha verificado que la sucesión efectivamente llega a 1 partiendo desde cualquier número natural menor que 2^{58} .

(a) Construya un programa que entregue la secuencia de Collatz de un número entero:

```
n: 18
18 9 28 14 7 22 11 34 17 52 26 13 40 20 10 5 16 8 4 2 1
```

```
n: 19
19 58 29 88 44 22 11 34 17 52 26 13 40 20 10 5 16 8 4 2 1
```

(b) Construya un programa que grafique los largos de las secuencias de Collatz de los números enteros positivos menores que el ingresado por el usuario:

```
n: 10
1 *
2 **
3 *****
4 ***
5 *****
6 *****
7 *****
8 ****
9 *****
10 *****
```

```
n: 16
1 *
2 **
3 *****
4 ***
5 *****
6 *****
7 *****
8 ****
9 *****
10 *****
11 *****
12 *****
13 *****
14 *****
15 *****
16 *****
```

17. En estadística descriptiva, se define el **rango** de un conjunto de datos reales como la diferencia entre el mayor y el menor de los datos. Por ejemplo, si los datos son: 5.96, 6.74, 7.43, 4.99, 7.20, 0.56 y 2.80 entonces el rango es $7.43 - 0.56 = 6.87$. Construya la función `rango(cantidad)` que reciba como parámetro la cantidad de números que el usuario ingresará, para luego pedirlos por pantalla y mostrar el rango. Suponga que todos los datos ingresados son válidos.

```
n = int(raw_input('Cantidad de valores: '))
r = rango(n)
print 'El rango es', r
```

```
Cantidad de valores: 8
Valor 1: 4.3
Valor 2: 0.76
Valor 3: 5.34
Valor 4: 1.23
Valor 5: 6.45
Valor 6: 0.66
Valor 7: 7.88
Valor 8: 3.45
El rango es 7.22
```

```
Cantidad de valores: 5
Valor 1: 23.12
Valor 2: 5.67
Valor 3: 3.52
Valor 4: 45.89
Valor 5: 123.56
El rango es 120.04
```


18. En finanzas, el valor actual neto es un indicador de la rentabilidad que tendrá un proyecto. Se calcula sumando los flujos de dinero de cada mes divididos por $(1 + r)^n$, donde n es el número del mes y r es la tasa de descuento mensual, y restando la inversión inicial.

Por ejemplo, en un proyecto en que la inversión inicial es \$900, los flujos de dinero estimados para los primeros cuatro meses son \$550, \$230, \$341 y \$190, y la tasa de descuento mensual es de 4 %, el valor actual neto es:

$$\text{VAN} = -900 + \frac{550}{(1 + 0.04)^1} + \frac{230}{(1 + 0.04)^2} + \frac{341}{(1 + 0.04)^3} + \frac{190}{(1 + 0.04)^4}$$

Si el VAN da negativo, entonces no es conveniente comenzar el proyecto. Construya un programa que pida al usuario ingresar la inversión inicial y el porcentaje de tasa de descuento. A continuación, debe preguntar el flujo de dinero estimado para cada mes y mostrar cuál es la parte entera del VAN hasta ese momento. El programa debe terminar apenas el VAN comience a dar positivo. Suponga que todos los datos ingresados son válidos.

```
Inversion inicial:900
% tasa de descuento:4
Flujo mes 1:550
VAN: -371
Flujo mes 2:230
VAN: -158
Flujo mes 3:341
VAN: 144
```

```
Inversion inicial:1500
% tasa de descuento:6
Flujo mes 1:450
VAN: -1075
Flujo mes 2:560
VAN: -577
Flujo mes 3:480
VAN: -174
Flujo mes 4:230
VAN: 8
```

```
Inversion inicial:500
% tasa de descuento:2
Flujo mes 1:430
VAN: -78
Flujo mes 2:120
VAN: 36
```