

## UVA 4: Ciclos (parte II)

cristobal.loyolam@usm.cl

3 de abril de 2023

- El plazo de SMOJ es este sábado 8 de abril a las 23:55.
- Esta semana no habrá control 😊
- Horario de consultas online: martes desde 17:10 a 18:20.
- Los videos de la UVA 5 saldrán este jueves 6 de abril.
- La próxima tarea se publicará el lunes 10 de abril.
- Recordar pasar asistencia.

# Contar pares v1

Se quiere implementar un programa que solicite el ingreso de  $n$  números, y finalmente imprima la cantidad de números pares ingresados.

```
1  n = int(input("Ingrese la cantidad de números: "))
2  pares = 0
3  i = 1
4  while i <= n:
5      x = int(input("Ingrese un número: "))
6      if x % 2 == 0:
7          pares += 1
8      i += 1
9  print("La cantidad de pares ingresados es", pares)
```

# Contar pares v2

Se quiere implementar un programa que solicite al usuario ingresar números indefinidamente, hasta que escriba -1. Finalmente se quiere imprimir la cantidad de números pares ingresados.

```
1     continuar = True
2     pares = 0
3     while continuar:
4         x = int(input("Ingrese un número: "))
5         if x == -1:
6             continuar = False
7         else:
8             if x % 2 == 0:
9                 pares += 1
10    print("La cantidad de pares ingresados es", pares)
```

Se desea sumar los números ingresados por el usuario, hasta que ingrese el valor 0. El siguiente código tiene un error, ¿cuál es?

```
1     continuar = True
2     suma = 0
3     while continuar:
4         n = int(input('Ingrese nro a sumar o 0 para detener: '))
5         if n == 0:
6             suma += n
7         else:
8             continuar = False
9     print('La suma es', suma)
```

Versión correcta del código anterior:

```
1     continuar = True
2     suma = 0
3     while continuar:
4         n = int(input('Ingrese nro a sumar o 0 para detener: '))
5         if n != 0:
6             suma += n
7         else:
8             continuar = False
9     print('La suma es', suma)
```

## Sumar números v2

Se desea sumar los números ingresados por el usuario, hasta que ingrese el valor 0 o la suma obtenida sea 100 o más. El siguiente código tiene dos errores, ¿cuáles son?

```
1     n = int(input("Ingrese nro a sumar o 0 para detener: "))
2     meta = False
3     suma = 0
4     while n != 0 or meta == False:
5         suma += n
6         if suma >= 100:
7             meta = True
8         n = int(input("Ingrese nro a sumar o 0 para detener: "))
9     print('La suma es', suma)
```

Versión correcta del código anterior:

```
1     n = int(input("Ingrese nro a sumar o 0 para detener: "))
2     meta = False
3     suma = 0
4     while n != 0 and meta == False:
5         suma += n
6         if suma >= 100:
7             meta = True
8         else:
9             n = int(input("Ingrese nro a sumar o 0 para detener: "))
10    print('La suma es', suma)
```



# Lanzamiento de dos dados

El siguiente código muestra todas las combinaciones posibles al lanzar 2 dados:

```
1     i = 1
2     j = 1
3     while i <= 6:
4         while j <= 6:
5             print(i, j)
6             j += 1
7         j = 1
8         i += 1
```

# Tablas de multiplicar

Se quiere implementar un programa que imprima las tablas de multiplicar (con los factores desde 1 hasta 10) para los primeros  $n$  números naturales, con  $n$  determinado por el usuario. Complete el siguiente programa para obtener lo solicitado:

```
1      n = int(input("Ingrese el valor de n: "))
2      m = 10
3      i = 1
4      j = 1
5      while i <= __:
6          while j <= __:
7              print(i, "x", j, "=", i*j)
8              j += __
9          j = __
10         i += __
```

Ingrese el valor de n: 2

```
1 x 1 = 1
1 x 2 = 2
1 x 3 = 3
1 x 4 = 4
1 x 5 = 5
1 x 6 = 6
1 x 7 = 7
1 x 8 = 8
1 x 9 = 9
1 x 10 = 10
2 x 1 = 2
2 x 2 = 4
2 x 3 = 6
2 x 4 = 8
2 x 5 = 10
2 x 6 = 12
2 x 7 = 14
2 x 8 = 16
2 x 9 = 18
2 x 10 = 20
```

Ejemplo para  
 $n=2$

# Tablas de multiplicar

Solución:

```
1  n = int(input("Ingrese el valor de n: "))
2  m = 10
3  i = 1
4  j = 1
5  while i <= n:
6      while j <= m:
7          print(i, "x", j, "=", i*j)
8          j += 1
9      j = 1
10     i += 1
```

Se quiere implementar un programa que pida un número al usuario y calcule su factorial (el factorial de un número es el producto de todos los enteros entre 1 y el propio número). Corrija el siguiente código para que permita realizar lo solicitado en el enunciado:

```
1 resultado = 0
2 num = int(input("Dime un número: "))
3 contador = 1
4 while contador < num:
5     resultado = resultado * contador
6     contador = contador - 1
7 print("El resultado es", resultado)
```

Versión correcta del código anterior:

```
1 resultado = 1
2 num = int(input("Dime un número: "))
3 contador = 1
4 while contador <= num:
5     resultado = resultado * contador
6     contador = contador + 1
7 print("El resultado es", resultado)
```

La función exponencial se puede aproximar mediante la siguiente serie de Taylor:

$$\begin{aligned}f(x) = e^x &= \sum_{i=0}^{\infty} \frac{x^i}{i!} = \frac{x^0}{0!} + \frac{x^1}{1!} + \frac{x^2}{2!} + \frac{x^3}{3!} + \frac{x^4}{4!} + \frac{x^5}{5!} + \dots \\&= 1 + x + \frac{x^2}{2} + \frac{x^3}{6} + \frac{x^4}{24} + \frac{x^5}{120} + \dots\end{aligned}$$

Se pide implementar un programa que calcule la aproximación de orden  $t$  de la exponencial de  $x$ , con  $t$  y  $x$  ingresados por el usuario.

# Serie de Taylor (función exponencial)

```
1  t = int(input("Ingrese el orden de la aproximación: "))
2  x = float(input("Ingrese x: "))
3  i = 1
4  suma = 1
5  while i <= t: # itera sobre cada término de la suma
6      factorial = 1
7      j = 1
8      while j <= i: # calcula el denominador de cada término
9          factorial = factorial * j
10         j += 1
11     suma += x**i / factorial
12     i += 1
13 print("La aproximación es", suma)
```