

Técnicas de Programación

Instituto de Formación Técnica Superior Nro. 11
Docente: Lic. Norberto A. Orlando



Laboratorio UNIDAD 3: Valores Booleanos, Ejecución Condicional, Bucles, Listas y su procesamiento, Operaciones Lógicas y de Bit a Bit

Ejercicio 1

¿Cuál es el resultado de las siguientes comparaciones?

- a) `2 == 2` → `true`
- b) `2 == 2.` → `true`
- c) `1 == 2` → `false`

Ejercicio 2

Usando uno de los operadores de comparación en Python, escribe un programa simple de dos líneas que tome el parámetro **n** como entrada, que es un entero, e imprime **False** si **n** es menor que 100, y **True** if **n** es mayor o igual que 100. No debes crear ningún bloque if

```
n = int(input("Ingresa un número: "))  
print(n >= 100)
```

Ejercicio 3

Se debe calcular el pago de un impuesto, denominado Impuesto Personal de Ingresos (IPI, para abreviar) utilizando la siguiente regla:

- si el ingreso del ciudadano no era superior a \$85528, el impuesto era igual al 18% del ingreso menos \$556,02 (esta fue la llamada exención fiscal).

- si el ingreso era superior a esta cantidad, el impuesto era igual a \$14839.02, más el 32% del excedente sobre 85528 pesos. (Formula = $(\text{valoringresado} - 85528) \times 0.32 + 14839.02$)
- Si el impuesto calculado es menor que cero, solo significa que no hay impuesto (el impuesto es igual a cero).

Tu tarea es escribir una calculadora de impuestos.

Debe aceptar un valor de punto flotante: el ingreso.

A continuación, debe imprimir el impuesto calculado, redondeado a pesos totales. Hay una función llamada round() que hará el redondeo

```
income = float(input("Introduce el ingreso anual: "))

if income < 85528:
    tax = income * 0.18 - 556.02
else:
    tax = (income - 85528) * 0.32 + 14839.02

if tax < 0.0:
    tax = 0.0

tax = round(tax, 0)
print("El impuesto es:", tax, "pesos")
```

Ejercicio 4

Como seguramente sabrás, debido a algunas razones astronómicas, el año puede ser bisiesto o común. Los primeros tienen una duración de 366 días, mientras que los últimos tienen una duración de 365 días.

Desde la introducción del calendario Gregoriano (en 1582), se utiliza la siguiente regla para determinar el tipo de año:

- si el número del año no es divisible entre cuatro, es un año común.
- de lo contrario, si el número del año no es divisible entre 100, es un año bisiesto.
- de lo contrario, si el número del año no es divisible entre 400, es un año común.
- de lo contrario, es un año bisiesto.

Observa el código en el editor - solo lee un número de año y debe completarse con las instrucciones que implementan la prueba que acabamos de describir.

El código debe mostrar uno de los dos mensajes posibles, que son **Año Bisiesto** o **Año Común**, según el valor ingresado.

Sería bueno verificar si el año ingresado cae en la era Gregoriana y emitir una advertencia de lo contrario: **No dentro del período del calendario Gregoriano**. Consejo: utiliza los operadores != y %.

```
year = int(input("Introduce un año: "))

if year < 1582:
    print("No esta dentro del período del calendario Gregoriano")
else:
    # Escribe el bloque if-elif-elif-else aquí.
```

```
year = int(input("Introduce un año: "))

if year < 1582:
    print("No esta dentro del período del calendario Gregoriano")
else:
    if year % 4 != 0:
        print("Año Común")
    elif year % 100 != 0:
        print("Año Bisiesto")
    elif year % 400 != 0:
        print("Año Común")
    else:
        print("Año Bisiesto")
```

Ejercicio 5

¿Cuál es el resultado del siguiente fragmento de código?

a)

```
x = 5
y = 10
z = 8

print(x > y)
print(y > z)
```

False

True

b)

```
x, y, z = 5, 10, 8

print(x > z)
print((y - 5) == x)
```

False

True

c)

```
x, y, z = 5, 10, 8
x, y, z = z, y, x

print(x > z)
print((y - 5) == x)
```

True

False

d)

```
x = 10

if x == 10:
    print(x == 10)
if x > 5:
    print(x > 5)
if x < 10:
    print(x < 10)
else:
    print("else")
```

True

True

else

e)

```
x = "1"

if x == 1:
    print("one")
elif x == "1":
    if int(x) > 1:
        print("two")
    elif int(x) < 1:
        print("three")
    else:
        print("four")
if int(x) == 1:
    print("five")
else:
    print("six")
```

four

five

f)

```
x = 1
y = 1.0
z = "1"

if x == y:
    print("one")
if y == int(z):
    print("two")
elif x == y:
    print("three")
else:
    print("four")
```

One

Two

Ejercicio 6

Un mago junior ha elegido un número secreto. Lo ha escondido en una variable llamada **secret_number**. Quiere que todos los que ejecutan su programa jueguen el juego Adivina el número secreto, y adivina qué número ha elegido para ellos. ¡Quiénes no adivinen el número quedarán atrapados en un bucle sin fin para siempre! Desafortunadamente, él no sabe cómo completar el código.

Tu tarea es ayudar al mago a completar el código en el editor de tal manera que el código:

- pedirá al usuario que ingrese un número entero;
- utilizará un bucle while;
- comprobará si el número ingresado por el usuario es el mismo que el número escogido por el mago. Si el número elegido por el usuario es diferente al número secreto del mago, el usuario debería ver el mensaje "**¡Ja, ja! ¡Estás atrapado en mi bucle!**" y se le solicitará que ingrese un número nuevamente. Si el número ingresado por el usuario coincide con el número escogido por el mago, el número debe imprimirse en la pantalla, y el mago debe decir las siguientes palabras: "**¡Bien hecho! Eres libre ahora.**"

INFO EXTRA

Por cierto, observa la función `print()`. La forma en que lo hemos utilizado aquí se llama impresión multilínea. Puedes utilizar comillas triples para imprimir cadenas en varias líneas para facilitar la lectura del texto o crear un diseño especial basado en texto. Experimenta con ello.

```
secret_number = 777

print(
    """
+=====+
| ¡Bienvenido a mi juego!   |
| Introduce un número entero |
| y adivina qué número he   |
| elegido para ti.         |
| ¿Cuál es el número secreto? |
+=====+
    """)
```

```
secret_number = 777

print(
    """
+=====+
| ¡Bienvenido a mi juego, muggle!  |
| Introduce un número entero       |
| y adivina qué número he         |
| elegido para ti.                 |
| ¿Cuál es el número secreto?     |
+=====+
    """)

user_number = int(input("Ingresa el número: "))

while user_number != secret_number:
    print("¡Ja, ja! ¡Estás atrapado en mi bucle!")
    user_number = int(input("Ingresa el número nuevamente: "))
print(secret_number, "¡Bien hecho! Eres libre ahora.")
```

Ejercicio 7

La instrucción **break** se implementa para salir/terminar un bucle.

Diseña un programa que use un bucle `while` y le pida continuamente al usuario que ingrese una palabra a menos que ingrese "**chupacabra**" como la palabra de output secreta, en cuyo caso el mensaje "**Has dejado el bucle con éxito.**" debe imprimirse en la pantalla y el bucle debe terminar.

No imprimas ninguna de las palabras ingresadas por el usuario. Utiliza el concepto de ejecución condicional y la sentencia `break`.

```
while True:
    # Escribir código en el cuerpo del bucle.
    # Colocar una sentencia condicional.
    # Salir del bucle aquí.
```

```
# Imprimir el mensaje.
```

```
while True:
    word = input("Estás atrapado en un bucle infinito!\nIngresa la palabra secreta para salir del bucle: ")
    if word == "chupacabra":
        break
print("¡Has dejado el bucle con éxito!")
```

Ejercicio 8

La sentencia **continue** se usa para omitir el bloque actual y avanzar a la siguiente iteración, sin ejecutar las sentencias dentro del bucle.

Se puede usar tanto con bucles while y for.

Tu tarea aquí es muy especial: ¡Debes diseñar un devorador de vocales! Escribe un programa que use:

- un bucle for
- el concepto de ejecución condicional (if-elif-else).
- la sentencia continue.

Tu programa debe:

- pedir al usuario que ingrese una palabra.
- utiliza `user_word = user_word.upper()` para convertir la palabra ingresada por el usuario a mayúsculas.
- utiliza la ejecución condicional y la instrucción continue para "devorar" las siguientes vocales A, E, I, O, U de la palabra ingresada.
- imprime las letras no consumidas en la pantalla, cada una de ellas en una línea separada

Ejemplo:

Entrada de muestra:	Salida esperada:
Gregory	G R G R Y

```
user_word = input("Ingresa tu palabra: ")
user_word = user_word.upper()

for letter in user_word:
    # if-elif-else block, sentencias continue anidadas
    # ...
    # ...
    # ...
    #else:
    #     print(letter)
```

```
user_word = input("Ingresa tu palabra: ")
user_word = user_word.upper()

for letter in user_word:
    if letter == "A":
        continue
    elif letter == "E":
        continue
```



```
elif letter == "I":
    continue
elif letter == "O":
    continue
elif letter == "U":
    continue
else:
    print(letter)
```

Ejercicio 9

Crea un bucle for que cuente de 0 a 10, e imprima números impares en la pantalla

```
for i in range(0, 11):
    if i % 2 != 0:
        print(i)
```

Ejercicio 10

Crea un bucle while que cuente de 0 a 10, e imprima números impares en la pantalla.

```
x = 1
while x < 11:
    if x % 2 != 0:
        print(x)
    x += 1
```

Ejercicio 11

Crea un programa con un bucle for y una sentencia break. El programa debe iterar sobre los caracteres en una dirección de correo electrónico, salir del bucle cuando llegue al símbolo @ e imprimir la parte antes de @ en una línea.

```
for ch in "Norberto.orlando@gmail.com":
    if ch == "@":
        break
    print(ch, end="")
```

Ejercicio 12

Crea un programa con un bucle for y una sentencia continue. El programa debe iterar sobre una cadena de dígitos, reemplazar cada 0 con x, e imprimir la cadena modificada en la pantalla.

```
for digit in "0165031806510":
    if digit == "0":
        print("x", end="")
        continue
    print(digit, end="")
```

Ejercicio 13

¿Cuál es el output de los siguientes códigos?

a)


```
n = 3

while n > 0:
    print(n + 1)
    n -= 1
else:
    print(n)
```

4
3
2
0

b)

```
n = range(4)

for num in n:
    print(num - 1)
else:
    print(num)
```

-1
0
1
2
3

c)

```
for i in range(0, 6, 3):
    print(i)
```

0
3

Ejercicio 14

¿Cuál es el resultado del siguiente fragmento de código?

```
1 | x = 1
2 | y = 0
3 |
4 | z = ((x == y) and (x == y)) or not(x == y)
5 | print(not(z))
6 |
```

False

Ejercicio 15

Ingresar el sueldo de una persona, si supera los 3000 dolares mostrar un mensaje en pantalla indicando que debe abonar impuestos.

```
suelo=int(input("Ingrese cual es su sueldo:"))
if sueldo>3000:
    print("Esta persona debe abonar impuestos")
```

Ejercicio 16

Realizar un programa que solicite ingresar dos números distintos y muestre por pantalla el mayor de ellos.

```
num1=int(input("Ingrese primer valor:"))
num2=int(input("ingrese segundo valor:"))
print("El valor mayor es")
if num1>num2:
    print(num1)
else:
    print(num2)
```

Ejercicio 17

Realizar un programa que solicite la carga por teclado de dos números, si el primero es mayor al segundo informar su suma y diferencia, en caso contrario informar el producto y la división del primero respecto al segundo.

```
num1=int(input("Ingrese primer valor:"))
num2=int(input("Ingrese segundo valor:"))
if num1>num2:
    suma=num1+num2
    print("La suma de los dos valores es")
    print(suma)
    resta=num1-num2
    print("La diferencia de los dos valores es")
    print(resta)
else:
    producto = num1*num2;
    print("El producto de los dos valores es")
    print(producto)
    division = num1/num2;
    print("La división de los dos valores es")
    print(division)
```

Ejercicio 18

Se ingresan tres notas de un alumno, si el promedio es mayor o igual a siete mostrar un mensaje "Promocionado".

```
nota1=int(input("Ingrese primer nota:"))
nota2=int(input("Ingrese segunda nota:"))
nota3=int(input("INGrese la tercer nota:"))
promedio=(nota1 + nota2 + nota3)/3
if promedio>=7:
    print("Promocionado")
```

Ejercicio 19

Se ingresa por teclado un número entero positivo de uno o dos dígitos (1..99) mostrar un mensaje indicando si el número tiene uno o dos dígitos.

(Tener en cuenta que condición debe cumplirse para tener dos dígitos un número entero)

```
num=int(input("Ingrese un valor entero de 1 o 2 dígitos:"))
if num<10:
    print("Tiene un dígito")
else:
    print("Tiene dos dígitos")
```

Ejercicio 20

Confeccionar un programa que pida por teclado tres notas de un alumno, calcule el promedio e imprima alguno de estos mensajes:

Si el promedio es ≥ 7 mostrar "Promocionado".

Si el promedio es ≥ 4 y < 7 mostrar "Regular".

Si el promedio es < 4 mostrar "Reprobado".

```
nota1=int(input("Ingrese primer nota:"))
nota2=int(input("Ingrese segunda nota:"))
nota3=int(input("Ingrese tercer nota:"))
prom=(nota1+nota2+nota3)/3
if prom>=7:
    print("Promocionado")
else:
    if prom>=4:
        print("Regular")
    else:
        print("Reprobado")
```

Ejercicio 21

Se cargan por teclado tres números distintos. Mostrar por pantalla el mayor de ellos.

```
num1=int(input("Ingrese primer valor:"))
num2=int(input("Ingrese segunda valor:"))
num3=int(input("Ingrese tercer valor:"))
if num1>num2:
    if num1>num3:
        print(num1)
    else:
        print(num3)
else:
    if num2>num3:
        print(num2)
    else:
        print(num3)
```

Ejercicio 22

Se ingresa por teclado un valor entero, mostrar una leyenda que indique si el número es positivo, negativo o nulo (es decir cero)

```
num=int(input("Ingrese un valor:"))
if num==0:
    print("Se ingresó el cero")
else:
    if num>0:
        print("Se ingresó un valor positivo")
    else:
```

```
print("Se ingresó un valor negativo")
```

Ejercicio 23

Confeccionar un programa que permita cargar un número entero positivo de hasta tres cifras y muestre un mensaje indicando si tiene 1, 2, o 3 cifras. Mostrar un mensaje de error si el número de cifras es mayor.

```
num=int(input("Ingrese un valor de hasta tres dígitos positivo:"))
if num<10:
    print("Tiene un dígito")
else:
    if num<100:
        print("Tiene dos dígitos")
    else:
        if num<1000:
            print("Tiene tres dígitos")
        else:
            print("Error en la entrada de datos.")
```

Ejercicio 24

Un postulante a un empleo, realiza un test de capacitación, se obtuvo la siguiente información: cantidad total de preguntas que se le realizaron y la cantidad de preguntas que contestó correctamente. Se pide confeccionar un programa que ingrese los dos datos por teclado e informe el nivel del mismo según el porcentaje de respuestas correctas que ha obtenido, y sabiendo que:

- Nivel máximo: Porcentaje \geq 90%.
- Nivel medio: Porcentaje \geq 75% y $<$ 90%.
- Nivel regular: Porcentaje \geq 50% y $<$ 75%.
- Fuera de nivel: Porcentaje $<$ 50%.

```
totalpreguntas=int(input("Ingrese la cantidad total de preguntas del examen:"))
totalcorrectas=int(input("Ingrese la cantidad total de preguntas contestadas correctamente:"))
porcentaje=totalcorrectas * 100 / totalpreguntas
if porcentaje $\geq$ 90:
    print("Nivel máximo")
else:
    if porcentaje $\geq$ 75:
        print("Nivel medio")
    else:
        if porcentaje $\geq$ 50:
            print("Nivel regular")
        else:
            print("Fuera de nivel")
```

Ejercicio 25

Se carga una fecha (día, mes y año) por teclado. Mostrar un mensaje si corresponde al primer trimestre del año (enero, febrero o marzo) Cargar por teclado el valor numérico del día, mes y año.

```
dia=int(input("Ingrese nro de día:"))
mes=int(input("Ingrese nro de mes:"))
año=int(input("Ingrese nro de año:"))
if mes==1 or mes==2 or mes==3:
    print("Corresponde al primer trimestre")
```

Ejercicio 26

Realizar un programa que pida cargar una fecha cualquiera, luego verificar si dicha fecha corresponde a Navidad.

```
dd=int(input("Ingrese nro de día:"))
mm=int(input("Ingrese nro de mes:"))
aa=int(input("Ingrese nro de año:"))
if mm==12 and dd==25:
    print("La fecha ingresada corresponde a navidad.")
```

Ejercicio 27

Se ingresan por teclado tres números, si todos los valores ingresados son menores a 10, imprimir en pantalla la leyenda "Todos los números son menores a diez".

```
num1=int(input("Ingrese primer valor:"))
num2=int(input("Ingrese segundo valor:"))
num3=int(input("Ingrese tercer valor:"))
if num1<10 and num2<10 and num3<10:
    print("Todos los números son menores a diez")
```

Ejercicio 28

Se ingresan por teclado tres números, si al menos uno de los valores ingresados es menor a 10, imprimir en pantalla la leyenda "Alguno de los números es menor a diez".

```
num1=int(input("Ingrese primer valor:"))
num2=int(input("Ingrese segundo valor:"))
num3=int(input("Ingrese tercer valor:"))
if num1<10 or num2<10 or num3<10:
    print("Alguno de los números es menor a diez")
```

Ejercicio 29

Escribir un programa que pida ingresar la coordenada de un punto en el plano, es decir dos valores enteros x e y (distintos a cero). Posteriormente imprimir en pantalla en que cuadrante se ubica dicho punto. (1º Cuadrante si $x > 0$ Y $y > 0$, 2º Cuadrante: $x < 0$ Y $y > 0$, etc.)

```
x=int(input("Ingrese coordenada x:"))
y=int(input("Ingrese coordenada y:"))
if x>0 and y>0:
    print("Se encuentra en el primer cuadrante")
else:
    if x<0 and y>0:
        print("Se encuentra en el segundo cuadrante")
    else:
```

```

if x<0 and y<0:
    print("Se encuentra en el tercer cuadrante")
else:
    print("Se encuentra en el cuarto cuadrante")

```

Ejercicio 30

De un operario se conoce su sueldo y los años de antigüedad. Se pide confeccionar un programa que lea los datos de entrada e informe:

- Si el sueldo es inferior a 500 y su antigüedad es igual o superior a 10 años, otorgarle un aumento del 20 %, mostrar el sueldo a pagar.
- Si el sueldo es inferior a 500 pero su antigüedad es menor a 10 años, otorgarle un aumento de 5 %.
- Si el sueldo es mayor o igual a 500 mostrar el sueldo en pantalla sin cambios.

```

sueldo=int(input("Ingrese sueldo del empleado:"))
antiguedad=int(input("Ingrese su antigüedad en años:"))
if sueldo<500 and antiguedad>10:
    aumento=sueldo*0.20
    sueldototal=sueldo+aumento
    print("Sueldo a pagar")
    print(sueldototal)
else:
    if sueldo<500:
        aumento=sueldo*0.05
        sueldototal=sueldo+aumento
        print("Sueldo a pagar")
        print(sueldototal)
    else:
        print("Sueldo a pagar")
        print(sueldo)

```

Ejercicio 31

Realizar un programa que imprima en pantalla los números del 1 al 100. (utilizando while)

```

x=1
while x<=100:
    print(x)
    x=x+1

```

Ejercicio 32

Codificar un programa que solicite la carga de un valor positivo y nos muestre desde 1 hasta el valor ingresado de uno en uno. Ejemplo: Si ingresamos 30 se debe mostrar en pantalla los números del 1 al 30.

```

n=int(input("Ingrese el valor final:"))
x=1
while x<=n:
    print(x)
    x=x+1

```

Ejercicio 33

Escribir un programa que solicite ingresar 10 notas de alumnos y nos informe cuántos tienen notas mayores o iguales a 7 y cuántos menores.

```
x=1
conta1=0
conta2=0
while x<=10:
    nota=int(input("Ingrese nota:"))
    if nota>=7:
        conta1=conta1+1
    else:
        conta2=conta2+1
    x=x+1
print("Cantidad de alumnos con notas mayores o iguales a 7")
print(conta1)
print("Cantidad de alumnos con notas menores a 7")
print(conta2)
```

Ejercicio 34

Se ingresan un conjunto de n alturas de personas por teclado. Mostrar la altura promedio de las personas.

```
n=int(input("Cuántas personas hay:"))
x=1
suma=0
while x<=n:
    altura=float(input("Ingrese la altura:"))
    suma=suma+altura
    x=x+1
promedio=suma/n
print("Altura promedio")
print(promedio)
```

Ejercicio 35

En una empresa trabajan n empleados cuyos sueldos oscilan entre \$100 y \$500, realizar un programa que lea los sueldos que cobra cada empleado e informe cuántos empleados cobran entre \$100 y \$300 y cuántos cobran más de \$300. Además el programa deberá informar el importe que gasta la empresa en sueldos al personal.

```
n=int(input("Cuántos empleados tiene la empresa:"))
x=1
conta1=0
conta2=0
gastos=0
while x<=n:
    sueldo=float(input("Ingrese el sueldo del empleado:"))
    if sueldo<=300:
        conta1=conta1+1
    else:
        conta2=conta2+1
    gastos=gastos+sueldo
    x=x+1
print("Cantidad de empleados que cobran entre $100 y $300")
print(conta1)
print("Cantidad de empleados que cobran más de $300")
print(conta2)
print("Importe que gasta la empresa en sueldos al personal")
print(gastos)
```



```

    x=x+1
print("Cantidad de empleados con sueldos entre 100 y 300")
print(conta1)
print("Cantidad de empleados con sueldos mayor a 300")
print(conta2)
print("Gastos total de la empresa en sueldos")
print(gastos)

```

Ejercicio 36

Desarrollar un programa que permita cargar n números enteros y luego nos informe cuántos valores fueron pares y cuántos impares.

Emplear el operador “%” en la condición de la estructura condicional (este operador retorna el resto de la división de dos valores, por ejemplo 11%2 retorna un 1, if valor%2==0:)

```

x=1
pares=0
impares=0
n=int(input("Cuántos números ingresará:"))
while x<=n:
    valor=int(input("Ingrese el valor:"))
    if valor%2==0:
        pares=pares+1
    else:
        impares=impares+1
    x=x+1
print("Cantidad de pares")
print(pares)
print("Cantidad de impares")
print(impares)

```

Ejercicio 37

Realizar un programa que imprima en pantalla los números del 0 al 100. Resolverlo empleando for.

```

for x in range(101):
    print(x)

```

Ejercicio 38

Realizar un programa que imprima en pantalla los números del 20 al 30. Resolverlo empleando for.

```

for x in range(20,31):
    print(x)

```

Ejercicio 39

Imprimir todos los números impares que hay entre 1 y 100. Resolverlo empleando for.

```

for x in range(1,100,2):
    print(x)

```

Ejercicio 40

Escribir un programa que lea 10 números enteros y luego muestre cuántos valores ingresados fueron múltiplos de 3 y cuántos de 5. Debemos tener en cuenta que hay números que son múltiplos de 3 y de 5 a la vez. Resolverlo empleando for.

```

mul3=0
mul5=0

```

```

for f in range(10):
    valor=int(input("Ingrese un valor:"))
    if valor%3==0:
        mul3=mul3+1
    if valor%5==0:
        mul5=mul5+1
print("Cantidad de valores ingresados múltiplos de 3")
print(mul3)
print("Cantidad de valores ingresados múltiplos de 5")
print(mul5)

```

Ejercicio 41

Confeccionar un programa que lea n pares de datos, cada par de datos corresponde a la medida de la base y la altura de un triángulo. Resolverlo empleando for.

El programa deberá informar:

- De cada triángulo la medida de su base, su altura y su superficie.
- La cantidad de triángulos cuya superficie es mayor a 12.

```

n=int(input("Cuantos triángulos procesará:"))
cantidad=0
for x in range(n):
    basetri=int(input("Ingrese el valor de la base:"))
    altura=int(input("Ingrese el valor de la altura:"))
    superficie=basetri*altura/2
    print("La superficie es")
    print(superficie)
    if superficie>12:
        cantidad=cantidad+1
print("La cantidad de triángulos con superficie superior a 12 son")
print(cantidad)

```

Ejercicio 42

Desarrollar un programa que solicite la carga de 10 números e imprima la suma de los últimos 5 valores ingresados.

```

suma=0
for f in range(10):
    valor=int(input("Ingrese un valor:"))
    if f>4:
        suma=suma+valor
print("La suma de los últimos 5 valores es")
print(suma)

```

Ejercicio 43

Desarrollar un programa que muestre la tabla de multiplicar del 5 (del 5 al 50). Resolverlo empleando for.

```

for f in range(5,51,5):
    print(f)

```

Ejercicio 44

Confeccionar un programa que permita ingresar un valor del 1 al 10 y nos muestre la tabla de multiplicar del mismo (los primeros 10 términos) Resolverlo empleando for.

Ejemplo: Si ingreso 3 deberá aparecer en pantalla los valores

```
3 x 1 = 3
3 x 2 = 6
3 x 3 = 9
3 x 4 = 12
3 x 5 = 15
3 x 6 = 18
3 x 7 = 21
3 x 8 = 24
3 x 9 = 27
3 x 10 = 30
```

```
valor=int(input("Ingrese un valor entre 1 y 10:"))
for f in range(1,11):
    tabla=valor*f
    print(str(valor) + ' x ' + str(f) + ' = ' + str(tabla))
```

Ejercicio 45

Realizar un programa que lea los lados de n triángulos, Resolverlo empleando for, e informar:

- a) De cada uno de ellos, qué tipo de triángulo es: equilátero (tres lados iguales), isósceles (dos lados iguales), o escaleno (ningún lado igual)
- b) Cantidad de triángulos de cada tipo.

```
cantEQ=0
cantIS=0
cantES=0
n=int(input("Ingrese la cantidad de triángulos:"))
for f in range(n):
    lado1=int(input("Ingrese lado 1:"))
    lado2=int(input("Ingrese lado 2:"))
    lado3=int(input("Ingrese lado 3:"))
    if lado1==lado2 and lado1==lado3:
        print("Es un triángulo equilátero.")
        cantEQ=cantEQ+1
    else:
        if lado1==lado2 or lado1==lado3 or lado2==lado3:
            print("Es un triángulo isósceles.")
            cantIS=cantIS+1
        else:
            print("Es un triángulo escaleno.")
            cantES=cantES+1

print("Cantidad de triángulos equiláteros:")
print(cantEQ)
print("Cantidad de triángulos isósceles:")
print(cantIS)
print("Cantidad de triángulos escalenos:")
print(cantES)
```

Ejercicio 46

Dada una lista de cinco números: 1, 2, 3, 4, y 5.

- a) escribir una línea de código que solicite al usuario que reemplace el número central en la lista con un número entero ingresado por el usuario (Paso 1)
- b) escribir una línea de código que elimine el último elemento de la lista (Paso 2)
- c) escribir una línea de código que imprima la longitud de la lista existente (Paso 3).

```
hat_list = [1, 2, 3, 4, 5] # Esta es una lista existente de números

# Paso 1: escribe una línea de código que solicite al usuario
# reemplazar el número de en medio con un número entero ingresado por el usuario.
hat_list[2] = int(input("Ingresa un número entero: "))
# Paso 2: escribe aquí una línea de código que elimine el último elemento de la lista.
del hat_list[-1]
# Paso 3: escribe aquí una línea de código que imprima la longitud de la lista
existente.
print(len(hat_list))
print(hat_list)
```

Ejercicio 47

¿Cuál es el resultado de los siguientes fragmentos de código?

a)

```
1 | lst = [1, 2, 3, 4, 5]
2 | lst.insert(1, 6)
3 | del lst[0]
4 | lst.append(1)
5 |
6 | print(lst)
7 |
```

[6, 2, 3, 4, 5, 1]

b)

```
1 | lst = [1, 2, 3, 4, 5]
2 | lst_2 = []
3 | add = 0
4 |
5 | for number in lst:
6 |     add += number
7 |     lst_2.append(add)
8 |
9 | print(lst_2)
10 |
```

```
[1, 3, 6, 10, 15]
```

c)

```
1 lst = [1, [2, 3], 4]
2 print(lst[1])
3 print(len(lst))
4
```

```
[2, 3]
```

```
3
```

Ejercicio 48

¿Cuál es el resultado de los siguientes fragmentos de código?

a)

```
1 lst = ["D", "F", "A", "Z"]
2 lst.sort()
3
4 print(lst)
5
```

```
['A', 'D', 'F', 'Z']
```

b)

```
1 a = 3
2 b = 1
3 c = 2
4
5 lst = [a, c, b]
6 lst.sort()
7
8 print(lst)
9
```

```
[1, 2, 3]
```

c)

```

1  a = "A"
2  b = "B"
3  c = "C"
4  d = " "
5
6  lst = [a, b, c, d]
7  lst.reverse()
8
9  print(lst)
10

```

```
[' ', 'C', 'B', 'A']
```

Ejercicio 49

Escribir un programa que elimine todas las repeticiones de números de la lista. El objetivo es tener una lista en la que todos los números aparezcan no más de una vez.

Nota: Asume que la lista original está ya dentro del código - no tienes que ingresarla desde el teclado.

Sugerencia: Te recomendamos que crees una nueva lista como área de trabajo temporal - no necesitas actualizar la lista actual.

```
my_list = [1, 2, 4, 4, 1, 4, 2, 6, 2, 9]
```

```

my_list = [1, 2, 4, 4, 1, 4, 2, 6, 2, 9]
new_list = []
for number in my_list: # Recorre todos los números de la lista original.
    if number not in new_list: # Si el número no aparece dentro de la nueva lista...
        new_list.append(number) # ...agregarlo aquí.
my_list = new_list[:] # Crea una copia de new_list.
print("La lista con elementos únicos:")
print(my_list)

```

Ejercicio 50

¿Cuál es el resultado del siguiente fragmento de código?

a)

```

1  list_1 = ["A", "B", "C"]
2  list_2 = list_1
3  list_3 = list_2
4
5  del list_1[0]
6  del list_2[0]
7
8  print(list_3)
9

```

```
['C']
```

b)

```
1 list_1 = ["A", "B", "C"]
2 list_2 = list_1
3 list_3 = list_2
4
5 del list_1[0]
6 del list_2
7
8 print(list_3)
9
```

```
['B', 'C']
```

c)

```
1 list_1 = ["A", "B", "C"]
2 list_2 = list_1
3 list_3 = list_2
4
5 del list_1[0]
6 del list_2[:]
7
8 print(list_3)
9
```

```
[]
```

d)

```
1 list_1 = ["A", "B", "C"]
2 list_2 = list_1[:]
3 list_3 = list_2[:]
4
5 del list_1[0]
6 del list_2[0]
7
8 print(list_3)
9
```

```
['A', 'B', 'C']
```

e) Inserta in o not in en lugar de ??? para que el código genere el resultado esperado.

```
1 my_list = [1, 2, "in", True, "ABC"]
2
3 print(1 ??? my_list) # output True
4 print("A" ??? my_list) # output True
5 print(3 ??? my_list) # output True
6 print(False ??? my_list) # output False
7
```



```
my_list = [1, 2, "in", True, "ABC"]

print(1 in my_list) # output True
print("A" not in my_list) # output True
print(3 not in my_list) # output True
print(False in my_list) # output False
```

Ejercicio 51

Definir una lista que almacene 5 enteros. Sumar todos sus elementos y mostrar dicha suma.

```
lista=[10,7,3,7,2]
suma=0
x=0
while x<len(lista):
    suma=suma+lista[x]
    x=x+1
print("Los elementos de la lista son")
print(lista)
print("La suma de todos sus elementos es")
print(suma)
```

Ejercicio 52

Definir una lista por asignación que almacene los nombres de los primeros cuatro meses de año.

Mostrar el primer y último elemento de la lista solamente.

```
meses=["enero", "febrero", "marzo", "abril"]
print(meses[0]) # se muestra enero
print(meses[3]) # se muestra abril
```

Ejercicio 53

Definir una lista por asignación que almacene en la primer componente el nombre de un alumno y en las dos siguientes sus notas. Imprimir luego el nombre y el promedio de las dos notas.

```
lista=["ana", 7, 9]
print("Nombre del alumno:")
print(lista[0])
promedio=(lista[1]+lista[2])/2
print("Promedio de sus dos notas:")
print(promedio)
```

Ejercicio 54

Definir por asignación una lista con 8 elementos enteros. Contar cuantos de dichos valores almacenan un valor superior a 100.

```
lista=[1000, 6000, 400, 23, 130, 400, 60, 2000]
cantidad=0
x=0
while x<len(lista):
    if lista[x]>100:
        cantidad+=1
    x+=1

print("La lista esta constituida por los elementos:")
```

```
print(lista)
print("La cantidad de valores mayores a 100 en la lista son:")
print(cantidad)
```

Ejercicio 55

Definir una lista vacía y luego solicitar la carga de 5 enteros por teclado y añadirlos a la lista. Imprimir la lista generada.

```
#definimos una lista vacia
lista=[]
#disponemos un ciclo de 5 vueltas
for x in range(5):
    valor=int(input("Ingrese un valor entero:"))
    lista.append(valor)

#imprimimos la lista
print(lista)
```

Ejercicio 56

Realizar la carga de valores enteros por teclado, almacenarlos en una lista. Finalizar la carga de enteros al ingresar el cero. Mostrar finalmente la lista y su tamaño.

```
lista=[]
valor=int(input("Ingresar valor (0 para finalizar):"))
while valor!=0:
    lista.append(valor)
    valor=int(input("Ingresar valor (0 para finalizar):"))

print("Tamano de la lista:")
print(len(lista))
```

Ejercicio 57

Almacenar en una lista los sueldos (valores float) de 5 operarios. Imprimir la lista y el promedio de sueldos.

```
suealdos=[]
suma=0
for x in range(5):
    valor=float(input("Ingrese el sueldo del operario:"))
    suealdos.append(valor)
    suma=suma+valor

print("Lista de suealdos")
print(suealdos)
promedio=suma/5
print("Promedio de suealdos")
print(promedio)
```

Ejercicio 58

Una empresa tiene dos turnos (mañana y tarde) en los que trabajan 8 empleados (4 por la mañana y 4 por la tarde) Confeccionar un programa que permita almacenar los sueldos de los empleados agrupados en dos listas.

Imprimir las dos listas de sueldos.

```
sueldosman=[]
print("Sueldos turno mañana")
for x in range(4):
    valor=float(input("Ingrese sueldo:"))
    sueldosman.append(valor)

sueldostar=[]
print("Sueldos turno tarde")
for x in range(4):
    valor=float(input("Ingrese sueldo:"))
    sueldostar.append(valor)

print("Turno mañana")
print(sueldosman)
print("Turno tarde")
print(sueldostar)
```

Ejercicio 59

Crear y cargar una lista con 5 enteros. Implementar un algoritmo que identifique el mayor valor de la lista.

```
lista=[]
for x in range(5):
    valor=int(input("Ingrese valor:"))
    lista.append(valor)

mayor=lista[0]
for x in range(1,5):
    if lista[x]>mayor:
        mayor=lista[x]

print("Lista completa")
print(lista)
print("Mayor de la lista")
print(mayor)
print(max(lista))
```

Ejercicio 60

Crear y cargar una lista con 5 enteros por teclado. Implementar un algoritmo que identifique el menor valor de la lista y la posición donde se encuentra.

```
lista=[]
for x in range(5):
    valor=int(input("Ingrese valor:"))
    lista.append(valor)
```

```

menor=lista[0]
posicion=0
for x in range(1,5):
    if lista[x]<menor:
        menor=lista[x]
        posicion=x

print("Lista completa")
print(lista)
print("Menor de la lista")
print(menor)
print(min(lista))
print("Posicion del menor en la lista")
print(posicion)

```

Ejercicio 61

Ingresar por teclado los nombres de 5 personas y almacenarlos en una lista. Mostrar el nombre de persona menor en orden alfabético.

```

nombres=[]
for x in range(5):
    nom=input("Ingrese nombre de persona:")
    nombres.append(nom)

nombremenor=nombres[0]
for x in range(1,5):
    if nombres[x]<nombremenor:
        nombremenor=nombres[x]

print("La lista completa de nombres ingresado son")
print(nombres)
print("El nombre menor en orden alfabetico es:")
print(nombremenor)

```

Ejercicio 62

Cargar una lista con 5 elementos enteros. Imprimir el mayor y un mensaje si se repite dentro de la lista (es decir si dicho valor se encuentra en 2 o más posiciones en la lista)

```

lista=[]
for x in range(5):
    valor=int(input("Ingrese valor:"))
    lista.append(valor)

mayor=lista[0]
for x in range(1,5):
    if lista[x]>mayor:
        mayor=lista[x]

print("Lista completa")
print(lista)

```

```

print("Mayor de la lista")
print(mayor)

# contamos cuantas veces se encuentra el mayor en la lista
cantidad=0
for x in range(5):
    if lista[x]==mayor:
        cantidad=cantidad+1
if cantidad>1:
    print("El mayor se repite en la lista", cantidad, "veces")

```

Ejercicio 63

Desarrollar un programa que permita cargar 5 nombres de personas y sus edades respectivas. Luego de realizar la carga por teclado de todos los datos imprimir los nombres de las personas mayores de edad (mayores o iguales a 18 años)

```

nombres=[]
edades=[]
for x in range(5):
    nom=input("Ingrese el nombre de la persona:")
    nombres.append(nom)
    ed=int(input("Ingrese la edad de dicha persona:"))
    edades.append(ed)

print("Nombre de las personas mayores de edad:")
for x in range(5):
    if edades[x]>=18:
        print(nombres[x], "tiene", edades[x])

```

Ejercicio 64

En un curso de 4 alumnos se registraron las notas de sus exámenes y se deben procesar de acuerdo a lo siguiente:

- Ingresar nombre y nota de cada alumno (almacenar los datos en dos listas paralelas)
- Realizar un listado que muestre los nombres, notas y condición del alumno. En la condición, colocar "Muy Bueno" si la nota es mayor o igual a 8, "Bueno" si la nota está entre 4 y 7, y colocar "Insuficiente" si la nota es inferior a 4.
- Imprimir cuantos alumnos tienen la leyenda "Muy Bueno".

```

nombres=[]
notas=[]
for x in range(4):
    nom=input("Ingrese nombre del alumno:")
    nombres.append(nom)
    no=int(input("Ingrese la nota de dicho alumno:"))
    notas.append(no)

cantidad=0
for x in range(4):
    if notas[x]>=8:
        print("Nombre:",nombres[x],"Nota:",notas[x],"Muy Bueno")
        cantidad+=1
    else:

```

```

    if notas[x]>=4:
        print("Nombre:",nombres[x],"Nota:",notas[x],"Bueno")
    else:
        print("Nombre:",nombres[x],"Nota:",notas[x],"Insuficiente")

print("La cantidad de alumnos muy buenos son", cantidad)

```

Ejercicio 65

Realizar un programa que pida la carga de dos listas numéricas enteras de 4 elementos cada una. Generar una tercer lista que surja de la suma de los elementos de la misma posición de cada lista. Mostrar esta tercer lista.

```

lista1=[]
print("Carga de la primer lista")
for x in range(4):
    valor=int(input("Ingrese valor:"))
    lista1.append(valor)

lista2=[]
print("Carga de la segunda lista")
for x in range(4):
    valor=int(input("Ingrese valor:"))
    lista2.append(valor)

listasuma=[]
for x in range(4):
    suma=lista1[x]+lista2[x]
    listasuma.append(suma)

print("Lista 1",lista1, "+", "Lista2", lista2, "=", listasuma)

```

Ejercicio 66

Se debe crear y cargar una lista donde almacenar 5 sueldos. Ordenar de menor a mayor la lista.

```

sueldos=[]
for x in range(5):
    valor=int(input("Ingrese sueldo:"))
    sueldos.append(valor)

print("Lista sin ordenar", sueldos)

for k in range(4):
    for x in range(4):
        if sueldos[x]>sueldos[x+1]:
            aux=sueldos[x]
            sueldos[x]=sueldos[x+1]
            sueldos[x+1]=aux

print("Lista ordenada",sueldos)

```

Ejercicio 68

Crear y cargar en un lista los nombres de 5 países y en otra lista paralela la cantidad de habitantes del mismo. Ordenar alfabéticamente e imprimir los resultados. Por último ordenar con respecto a la cantidad de habitantes (de mayor a menor) e imprimir nuevamente.

```
países=[]
habitantes=[]
for x in range(5):
    nom=input("Ingrese el nombre del país:")
    países.append(nom)
    cant=int(input("Cantidad de habitantes"))
    habitantes.append(cant)

# ordenamiento alfabético
for k in range(4):
    for x in range(4-k):
        if países[x]>países[x+1]:
            aux1=países[x]
            países[x]=países[x+1]
            países[x+1]=aux1
            aux2=habitantes[x]
            habitantes[x]=habitantes[x+1]
            habitantes[x+1]=aux2

print("Listado de países en orden alfabético")
for x in range(5):
    print(países[x],habitantes[x])

# ordenamiento por cantidad de habitantes
for k in range(4):
    for x in range(4-k):
        if habitantes[x]<habitantes[x+1]:
            aux1=países[x]
            países[x]=países[x+1]
            países[x+1]=aux1
            aux2=habitantes[x]
            habitantes[x]=habitantes[x+1]
            habitantes[x+1]=aux2

print("Listado de países por cantidad de habitantes")
for x in range(5):
    print(países[x],habitantes[x])
```