

1. Un año bisiesto es aquel divisible por 4, excepto aquellos divisibles por 100 pero no por 400. Escribe un programa que solicite al usuario un año y determine si es bisiesto o no, y que finalice el programa hasta que el usuario coloque "Salir" o el número 0

```
def anio_bisiesto():
    # Solicito que se ingrese un año para ingresar al bucle
    anio = int(input("Ingrese un año para comprobar si es bisiesto o no: "))
    # Mientras el año no sea 0 pide que se ingrese otro año
    while anio != 0:
        # Un año bisiesto es divisible por 4
        if anio % 4 == 0:
            # Un año que da comienzo a los centenarios es bisiesto si es
            # divisible por 100 y por 400
            if anio % 100 == 0 and anio % 400 == 0:
                print("Bisiesto")
            else:
                print("No es bisiesto")
        else:
            print("Es bisiesto")
        # Solicito que ingrese nuevamente un año para comprobar si es o no
        # bisiesto
        anio = int(input("Ingrese un año para comprobar si es bisiesto o no: "))

# Ejecuto el programa
anio_bisiesto()
```

2. En una tienda, hay descuentos en función del total de compra. Escribe un programa que solicite al usuario el total de la compra y aplique automáticamente el descuento correspondiente según la siguiente tabla: 10% de descuento para compras mayores o iguales a \$100, 20% para compras mayores o iguales a \$200, y 30% para compras mayores o iguales a \$300. Que finalice el programa hasta que el usuario coloque "Salir" o el número 0.

```
def descuentos():
    # Solicito un primer valor para ingresar al bucle
    valor = input("Ingrese el precio: $ ")

    # Bucle
    while True:
        # Evaluo si la variable valor contiene la palabra "salir" o el
        # numero 0, en caso de contenerlo salgo del bucle
        if str(valor.lower()) == "salir" or int(valor) == 0:
            break
```

```

        # Aplico los descuentos correspondientes e imprimo el precio con el
        descuento por pantalla
        if int(valor) >= 300:
            descuento = 0.7
            precio = int(valor) * descuento
            print(f'El precio es de $ {valor}.\nDescuento aplicado: 30 % (-
            ${precio-descuento})\nPrecio final con descuento: ${precio}')
        elif int(valor) >= 200:
            descuento = 0.8
            precio = int(valor) * descuento
            print(f'El precio es de $ {valor}.\nDescuento aplicado: 20 % (-
            ${precio-descuento})\nPrecio final con descuento: ${precio}')
        elif int(valor) >= 100:
            descuento = 0.1
            precio = int(valor) * descuento
            print(f'El precio es de $ {valor}.\nDescuento aplicado: 10 %(-
            ${precio-descuento})\nPrecio final con descuento: ${precio}')

        # En caso de no aplicarse ningun descuento muestro el precio
        original
        else:
            print(f'El precio es de $ {valor}. No se realizan descuentos')

        # Solicito que se ingrese un nuevo precio o se escriba la palabra
        "salir"
        valor = input("Ingrese el precio: $ ")

# Ejecuto el programa
descuentos()

```

3. Considere que necesitamos un programa que realice conversiones entre diferentes unidades de medida. El programa debe solicitar al usuario una cantidad y una unidad de medida (metros, centímetros o pulgadas), y luego realizar la conversión a las otras dos unidades de medida.

El programa debe ejecutarse en un bucle continuo hasta que el usuario ingrese el valor 0 como cantidad. Cada vez que se solicite una cantidad y una unidad de medida, el programa debe mostrar los resultados de las conversiones a metros, centímetros y pulgadas. A continuación, se muestra un ejemplo de la interacción esperada:

```

Ingrese la cantidad (ingrese 0 para salir): 2.5
Ingrese la unidad de medida (metros, centimetros, pulgadas): centimetros
2.5 centimetros equivale a:
0.025 metros
2.5 centimetros
0.984251968503937 pulgadas
-----
Ingrese la cantidad (ingrese 0 para salir): 10
Ingrese la unidad de medida (metros, centimetros, pulgadas): pulgadas
10 pulgadas equivale a:
0.254 metros
25.4 centimetros
10 pulgadas
-----
Ingrese la cantidad (ingrese 0 para salir): 0

```

El programa debe realizar las conversiones según las siguientes relaciones:

- 1 metro equivale a 100 centímetros y a aproximadamente 39.37 pulgadas.
- 1 centímetro equivale a 0.01 metros y a aproximadamente 0.3937 pulgadas.
- 1 pulgada equivale a aproximadamente 0.0254 metros y a 2.54. centímetros

```

# Constantes
# Metro
M_A_CM = 100
M_A_IN = 39.97

# Centimetro
CM_A_M = 0.010
CM_A_IN = 0.3937

# Pulgadas
IN_A_M = 0.0254
IN_A_CM = 2.54

# Creo una función para convertir unidades
def conversor():
    # Solicito una medida al usuario
    medida = float(input("Indique la medida (ingrese 0 para salir): "))

    # Si el usuario ingresa 0 en la medida el programa sale
    if medida == 0:
        return False

```

```

# Solicito que se ingrese una unidad de medición
unidad = str(input("Ingrese la unidad de medición: "))

# Ingreso al bucle
while True:
    print("-----")

    # Conversiones e impresión de resultados
    if unidad.lower() == "metros":
        metros = medida
        centimetros = medida * M_A_CM
        pulgadas = medida * M_A_IN
        print(f'{medida} {unidad} equivalen a :')
        print(f'{metros} metros')
        print(f'{centimetros} centimetros')
        print(f'{pulgadas} pulgadas')
    elif unidad.lower() == "centimetros":
        metros = medida * CM_A_M
        centimetros = medida
        pulgadas = medida * CM_A_IN
        print(f'{medida} {unidad} equivalen a :')
        print(f'{metros} metros')
        print(f'{centimetros} centimetros')
        print(f'{pulgadas} pulgadas')
    elif unidad.lower() == "pulgadas":
        metros = medida * IN_A_M
        centimetros = medida * IN_A_CM
        pulgadas = medida
        print(f'{medida} {unidad} equivalen a :')
        print(f'{metros} metros')
        print(f'{centimetros} centimetros')
        print(f'{pulgadas} pulgadas')

    # Si algun dato es erróneo se solicita volver a ingresar
    else:
        medida = float(input("Datos incorrectos. Indique la medida (ingrese 0 para salir): "))
        unidad = str(input("Datos incorrectos. Ingrese la unidad de medición: "))
        print("-----")

    # Solicito nuevamente el ingreso de la medida y de la unidad
    medida = float(input("Indique la medida (ingrese 0 para salir): "))
    if medida == 0:
        break

```

```
unidad = str(input("Ingrese la unidad de medición: "))
```

```
# Ejecuto el programa  
conversor()
```

4. Conversor de sistema decimal a binario: Introducción: Escribe un programa que solicite al usuario un número en sistema decimal y lo convierta a su representación en sistema binario.

```
# Defino el programa  
def decimal_a_binario():  
    # Solicito un numero decimal para convertir a binario  
    decimal = int(input("Ingrese un número en decimal para convertir a  
binario: "))  
  
    # Cadena vacía donde se contendrá el binario del decimal ingresado  
    binario = ""  
  
    # Ingreso al bucle donde se realizaran divisiones sucesivas por el  
numero 2  
    while decimal // 2 > 0:  
        # Guardo el resto de la división del decimal en la cadena "binario"  
        binario += str(decimal % 2)  
        # Divido repetidamete el decimal por 2  
        decimal //= 2  
  
    # Dado que un binario siempre debe comenzar por 1, le agrego el 1  
faltante  
    binario += "1"  
  
    # La cadena creada contiene el binario, pero en sentido inverso, por lo  
cual la debo invertir  
    print(binario[::-1])  
  
# Ejecuto el programa  
decimal_a_binario()
```

5. Escribe un programa que elija aleatoriamente una palabra de una lista de palabras predefinidas. Luego, permite al usuario adivinar las letras de la palabra en un juego de ahorcado. El programa debe mostrar en pantalla el estado actual del juego (letras adivinadas y letras por adivinar) y permitir al usuario ingresar una letra en cada intento.

```
# Importo la biblioteca random  
import random  
  
# Elijo una palabra de una lista predefinida  
def palabra_definidas():
```

```

    palabras =
["Licenciatura","en","Gestión","de","Tecnologías","la","información"]

    # Elijo una palabra al azar de la lista
    al_azar = random.choice(palabras).lower()

    # Devuelvo una palabra y la misma palabra pero separada por letra
    return al_azar,list(al_azar)

# Programa
def adivinar(palabra_al_azar):
    # Tomo una palabra al azar que será utilizada para el ahorcado y una
    lista con las letras
    palabra_a_adivinar,lista_de_letras = palabra_al_azar

    # Creo una lista donde irán las letras adivinadas y una lista con
    guiones de igual longitud que la palabra buscada
    adivinadas = ["_"] * len(palabra_a_adivinar)
    # Imprimo la cantidad de letras a adivinar
    print(f'Cantidad de letras a adivinar:
{len(palabra_a_adivinar)}\n{"_"*len(palabra_a_adivinar)}\n')

    # Ciclo donde solicito letras y compruebo si están o no en el juego
    while True:
        # Solicito que se ingrese una letra, automáticamente el valor
        ingresado pasa a ser un string
        letra = str(input("Ingrese una letra: "))

        # Recorro la lista con las letras que contiene la palabra a
        descubrir
        for i in range(0,len(lista_de_letras)):

            # Comparo la i-ésima letra con la letra ingresada para ver si
            son iguales
            if lista_de_letras[i] == letra:
                # En caso que la letra ingresada esté en la palabra a
                descubrir, entonces reemplazo el guión bajo por la letra descubierta
                adivinadas[i] = letra

            # Creo una cadena, que une las letras descubiertas con las que están
            sin descubrir
            cadena = "".join(f"{str(adivinadas[i])}" for i in
range(0,len(adivinadas)))

            # Imprimo la cadena con las letras descubiertas hasta ahora en
            pantalla

```

```

        print(f'Letras adivinadas:\t\t{cadena}')

        # Si la palabra fue descubierta aparece un mensaje revelando la
palabra a descubrir
        if adivinadas == lista_de_letras:
            print(f'Adivinaste la palabra era:\t{palabra_a_adivinar}')
            # Finalizo el programa
            return False

# Programa principal
def main():
    adivinar(palabra_definidas())

# Ejecuto el programa
main()

```

6. Mencione las diferencias entre if, elif, else, while, y for.

IF: es una función condicional que en español equivale a la palabra “sí”. La función IF solicita un parámetro, el cual será evaluado y según el resultado continuará con los argumentos contenidos en su interior o no.

ELIF: es un comando que se utiliza dentro de la función IF. Sería una condición secundaria que también evalúa un argumento en caso de que la condición anterior (o anteriores).

Cumple la función de ser un comando condicional dentro de otro comando condicional.

ELSE: el comando ELSE equivale en español a “sino”. Es decir, si las condiciones brindadas en el IF no se cumplen, entonces se ejecutarán los argumentos contenidos dentro del ELSE.

WHILE: Es un bucle que contiene una condición para su ejecución, siempre que esa condición sea verdadera el programa o función se mantendrá del de este ciclo, repitiendo los comandos que contenga en su interior.

FOR: Es otra función de tipo cíclica. A diferencia de WHILE necesita de ciertos parámetros para su ejecución, como una variable y otro parámetro que determine su final (aunque también se puede indicar su comienzo, final y el salto que debe dar entre estos valores).