**Algoritmos y Programación**

**Unidad 4**

# Ejercicio 4.1. Escribir dos funciones que resuelvan los siguientes problemas:

# a) Dado un número entero 𝑛, indicar si es par o no.

def numero\_par (n):

“””

Devuelve si un número es par o impar.

Parameters:

int(n): Número entero.

Returns:

(str) Cadena de texto que indica “Su número es par o impar”.

“””

assert isinstance (n, int),“Su número ingresado debe ser un entero.”

if n % 2 == 0:

return(“Su número es par”) # Se podría agregar un print de ser necesario

else:

return(“Su número es impar”)

# b) Dado un número entero 𝑛, indicar si es primo o no.

def numero\_primo (n):

“””

Devuelve si un número es primo o no.

Parameters:

int(n): Número entero mayor a 0.

Returns:

(str) Cadena de texto que indica “Su número es primo o no es primo”.

“””

assert n > 0, “Su número año debe ser mayor a 0.”

assert isinstance (n, int), “Su número debe ser un número entero.”

if n <2:

return “Su número no es primo.”

for i in range (2,n):

if n % i == 0:

return “Su número no es primo.”

else:

return “Su número es primo.”

# Ejercicio 4.2. Escribir una implementación propia de la función abs, que devuelva el valor absoluto de cualquier valor que reciba.

print(abs(n))

En función sino sería:

def absoluto(n):

“””

Devuelve el valor absoluto del número entero o flotante.

Parameters:

int(n) or float(n): Número entero o flotante.

Returns:

(int) Devuelve el valor absoluto de ese número.

“””

assert isinstance (n, int) or (n,float), “Su número debe ser un número entero o flotante.” #

if n >= 0:

return n

else:

return -n

# Ejercicio 4.5. Escribir funciones que resuelvan los siguientes problemas:

# Dado un año indicar si es bisiesto. Nota: un año es bisiesto si es un número divisible por 4, pero no si es divisible por 100, excepto que también sea divisible por 400.

def bisiesto (n):

“””

Devuelve si un año es bisiesto.

Parameters:

int (n) Año ingresado en números enteros.

Returns:

(str) Cadena de texto que indica “Su año es o no es bisiesto.”

“””

assert isinstance (n, int), “Su año debe ser ingresado en números enteros.”

if n % 4 == 0 and not n % 100 == 0 or n % 4 == 0:

return “Su año es bisiesto.”

else:

return “Su año no es bisiesto.”

# Dado un mes y un año, devolver la cantidad de días correspondientes.

def dias(m,a):

“””

Devuelve la cantidad de días correspondientes a un mes y año/s previamente dados.

Parameters:

int (m,a) Mes y año ingresado en números enteros.

Returns:

(str) Cadena de texto que indica “La cantidad de (d) día/s es el equivalente al año y mes que ingresó.”

“””

assert isinstance (m, int), “Su mes debe ser ingresado en números enteros.”

assert isinstance (a, int), “Su año debe ser ingresado en números enteros.”

a = a \* 365 + (a // 4) # Le sumo un día cada 4 x bisiesto.

a = a

d\_meses = [0,31,28,31,30,31,30,31,31,30,31,30,31] # Para que sea más exacto recorro la duración de los meses hasta el dado y sumo su cantidad de días.

suma\_meses = 0

for i in range(1,m+1):

i = d\_meses[i] # i va a ir tomando el valor del array en la posición del mes.

suma\_meses = suma\_meses + i

m = suma\_meses

d = a + m # días = años + meses convertidos

return f “La cantidad de {d} día/s es el equivalente al año y mes que ingresó.”

#Ejemplo:

m = int(input(“Ingrese la cantidad de meses que quiere pasar a días:”))

a = int(input(“Ingrese la cantidad de años que quiere pasar a días:”))

print(días(m,a))

# Dada una fecha (día, mes, año), indicar si es válida o no.

def bisiesto (n):

“””

Devuelve si un año es bisiesto.

Parameters:

int (n) Año ingresado en números enteros.

Returns:

(bool) Indica si su año es bisiesto, devolviendo: Verdadero o Falso.

“””

assert isinstance (n, int), “Su año debe ser ingresado en números enteros.”

if n % 4 == 0 and not n % 100 == 0 or n % 4 == 0:

return True

else:

return False

def fecha(d,m,a):

“””

Devuelve si una fecha es válida tras recibirla previamente como día, mes y año en forma numérica.

Parameters:

int (d,m,a) Día, mes y año (se toma como válido hasta el año 2020).

Returns:

(str) Cadena de texto que indica “Su fecha (d,m,a) es válida o inválida”.

“””

assert isinstance (d, int), “Sus día/s deben ser ingresados en números enteros.”

assert isinstance (m, int), “Sus mes/es deben ser ingresados en números enteros.”

assert isinstance (a, int), ““Su año debe ser ingresado en números enteros.”

ano = bisiesto(a) #Le paso esta función

if ano == True and d >= 1 and d <= 31 and m >= 1 and m <= 12 and a >= 1 and a >= 2020:

return f “Su fecha {d,m,a} es válida.”

elif ano == False:

if d >= 29 and m == 2:

return f “Su fecha {d,m,a} es inválida.”

elif d >= 1 and d <= 31 and m >= 1 and m <= 12 and a >= 1 and a >= 2020:

return f “Su fecha {d,m,a} es válida.”

else:

return f “Su fecha {d,m,a} es inválida.”

else:

return f “Su fecha {d,m,a} es inválida.”

#Ejemplo:

print(fecha(d,m,a))

# Dada una fecha, indicar los días que faltan hasta fin de mes.

def bisiesto (n):

“””

Devuelve si un año es bisiesto.

Parameters:

int (n) Año ingresado en números enteros.

Returns:

(bool) Indica si su año es bisiesto, devolviendo: Verdadero o Falso.

“””

assert isinstance (n, int), “Su año debe ser ingresado en números enteros.”

if n % 4 == 0 and not n % 100 == 0 or n % 4 == 0:

return True

else:

return False

def d\_fin\_mes(d,m,a):

“””

Devuelve la cantidad de días que faltan hasta fin de mes, partiendo de una fecha dada.

Parameters:

int (d,m,a) Día, mes y año ingresados en números enteros.

Returns:

(str) Cadena de texto que indica “Faltan (d) día/s para llegar a fin de mes.”

“””

def feb(a):

“””

Devuelve la cantidad de días que tiene Febrero, dependiendo de si el año es o no bisiesto.

Parameters:

int (a) Año ingresados en números enteros.

Returns:

int (feb) La cantidad de días que tiene Febrero en ese año.

“””

assert isinstance (a, int), “Su año debe ser ingresado en números enteros.”

a\_bisiesto = bisiesto(a) # Llama a esa función.

if a\_bisiesto == True:

feb = 29

return feb

else:

feb = 28

return feb

assert isinstance (d, int), “Sus día/s deben ser ingresados en números enteros.”

assert isinstance (m, int), “Sus mes/es deben ser ingresados en números enteros.”

assert isinstance (a, int), ““Su año debe ser ingresado en números enteros.”

m\_31d = [1,3,5,7,8,10,12] # Meses con 31 días.

m\_30d = [4,6,9,11] # Meses con 30 días.

feb = feb (a) # Varía si es bisiesto, llama a esa función.

for i in range(len(m\_31d)):

i = m\_31d[i] # i va a ir tomando el valor del array para compararlo luego con el mes dado.

if i == m:

d\_restantes = 31 - d

return f “Faltan {d\_restantes} día/s para llegar al fin de mes que ingresó.”

for i in range(len(m\_30d)):

i = m\_30d[i] # i va a ir tomando el valor del array para compararlo luego con el mes dado.

if i == m:

d\_restantes = 30 - d

return f “Faltan {d\_restantes} día/s para llegar al fin de mes que ingresó.”

else:

d\_restantes = feb – d # Llamo a febrero que va a ser 28 o 29 según sea o no bisiesto.

return f “Faltan {d\_restantes} día/s para llegar al fin de mes que ingresó.”

#Ejemplo:

d = int(input(“Ingrese la cantidad de días de su fecha, le diremos cuantos días faltan para fin de mes:”))

m = int(input(“Ingrese la cantidad de meses de su fecha:”))

a = int(input(“Ingrese la cantidad de años de su fecha:”))

print(d\_fin\_mes(d,m,a))

# e) Dada una fecha, indicar los días que faltan hasta fin de año.

<https://github.com/Plofi/Tareas/blob/master/Clase%204/Algoritmos%20y%20Programaci%C3%B3n%20-%20Unidad%204/AyP_Unidad%204_Ej%204.5.e_Def%20dias_fin_a%C3%B1o.py#L109>

# f) Dada una fecha, indicar la cantidad de días transcurridos en ese año hasta esa fecha.

**Opción 1:**

<https://github.com/Plofi/Tareas/blob/master/Clase%204/Algoritmos%20y%20Programaci%C3%B3n%20-%20Unidad%204/AyP_Unidad%204_Ej%204.5.f_Def%20dias_transcurridos.py>

**Opción 2:**

<https://github.com/Plofi/Tareas/blob/master/Clase%204/Algoritmos%20y%20Programaci%C3%B3n%20-%20Unidad%204/AyP_Unidad%204_Ej%204.5.f_Def%20dias_transcurridos_op%202.py>

# g) Dadas dos fechas (día1, mes1, año1, día2, mes2, año2), indicar el tiempo transcurrido entre ambas, en años, meses y días. Nota: en todos los casos, invocar las funciones escritas previamente cuando sea posible.

<https://github.com/Plofi/Tareas/blob/master/Clase%204/Algoritmos%20y%20Programaci%C3%B3n%20-%20Unidad%204/AyP_Unidad%204_Ej%204.5.g_Def%20dma_entre_fechas.py>

# h) Dadas dos fechas (día1, mes1, año1, día2, mes2, año2), indicar el tiempo transcurrido entre ambas, en días.

<https://github.com/Plofi/Tareas/blob/master/Clase%204/Algoritmos%20y%20Programaci%C3%B3n%20-%20Unidad%204/AyP_Unidad%204_Ej%204.5.h_Def%20dias_entre_fechas.py>

# Ejercicio 4.6. Suponiendo que el primer día del año fue lunes, escribir una función que reciba un número con el día del año (de 1 a 366) y devuelva el día de la semana que le toca. Por ejemplo: si recibe '3' debe devolver 'miércoles', si recibe '9' debe devolver 'martes'.

<https://github.com/Plofi/Tareas/blob/master/Clase%204/Algoritmos%20y%20Programaci%C3%B3n%20-%20Unidad%204/AyP_Unidad%204_Ej%204.6_Def%20dias_semana.py>

# Ejercicio 4.7. Escribir un programa que reciba como entrada un entero representando un año (por ejemplo 751, 1999, o 2158), y muestre por pantalla el mismo año escrito en números romanos.

<https://github.com/Plofi/Tareas/blob/master/Clase%204/Algoritmos%20y%20Programaci%C3%B3n%20-%20Unidad%204/AyP_Unidad%204_Ej%204.7_Def%20num_romanos.py>

# Ejercicio 4.8. Programa de astrología: el usuario debe ingresar el día y mes de su cumpleaños y el programa le debe decir a qué signo corresponde. Aries: 21 de marzo al 20 de abril. Tauro: 21 de abril al 20 de mayo. Géminis: 21 de mayo al 21 de junio. Cáncer: 22 de junio al 23 de julio. Leo: 24 de julio al 23 de agosto. Virgo: 24 de agosto al 23 de septiembre. Libra: 24 de septiembre al 22 de octubre. Escorpio: 23 de octubre al 22 de noviembre. Sagitario: 23 de noviembre al 21 de diciembre. Capricornio: 22 de diciembre al 20 de enero. Acuario: 21 de enero al 19 de febrero. Piscis: 20 de febrero al 20 de marzo.

<https://github.com/Plofi/Tareas/blob/master/Clase%204/Algoritmos%20y%20Programaci%C3%B3n%20-%20Unidad%204/AyP_Unidad%204_Ej%204.8_Def%20signos.py>