

## Boletín 6

### Ejercicio 1

#### Código

```
Ejercicio01.py > ...
1
2 '''Crea un programa que pida dos números enteros al usuario y diga si alguno de ellos es
3 múltiplo del otro. Crea una función EsMultiplo que reciba los dos números, y devuelve si el
4 primero es múltiplo del segundo'''
5
6 #Método que comprueba que el dato introducido sea un número
7 #En caso de no serlo, se volverá a pedir el dato hasta que éste sea un valor numérico
8 def comprobarNumero(numero):
9     correcto = False
10    while correcto == False:
11        if numero.isdigit():
12            correcto = True
13        else:
14            print("ERROR -> Solo se permiten números enteros")
15            numero = input("Introduce un numero: ")
16
17 #Método que pide dos datos por parámetros y comprueba si el primero es múltiplo del segundo
18 def esMultiplo(num1, num2):
19     if num1 % num2 == 0:
20         print("El número", num1, "es múltiplo de", num2)
21     else:
22         print("El número", num1, "no es múltiplo de", num2)
23
24 #Se pide al usuario dos números y se comprueba que sea un dato numérico
25 numero1 = input("Introduce un numero: ")
26 comprobarNumero(numero1)
27 numero2 = input("introduce otro número: ")
28 comprobarNumero(numero2)
29
30 #Se comprueba si el primer dato es múltiplo del segundo
31 esMultiplo(int(numero1), int(numero2))
```

#### Ejecución

```
Introduce un numero: 24
introduce otro número: 8
El número 24 es múltiplo de 8
```

```
Introduce un numero: 25
introduce otro número: 4
El número 25 no es múltiplo de 4
```

## Código

```
Ejercicio02.py > ...
1  '''Crea una función "calcularMaxMin" que recibe una lista con valores numéricos y devuelve el
2  valor máximo y el mínimo. Crea un programa que pida números por teclado y muestre el
3  máximo y el mínimo, utilizando la función anterior'''
4
5  #Método que comprueba que el dato introducido sea un número
6  #En caso de no serlo, se volverá a pedir el dato hasta que éste sea un valor numérico
7  def comprobarNumero(numero):
8      correcto = False
9      while correcto == False:
10         if numero.isdigit():
11             correcto = True
12         else:
13             print("ERROR -> Solo se permiten números enteros")
14             numero = input("Introduce un numero: ")
15
16  #Método que ordena una lista y muestra el valor máximo y el mínimo
17  def calcularMaxMin(lista):
18      lista.sort()
19      print("El valor máximo es", lista[len(lista) - 1])
20      print("El valor mínimo es", lista[0])
21
22  #Se crea una lista y un boolean iniciándolo a False
23  lista = []
24  salir = False
25
26  #Se pide un número y se introduce en una lista. Se da la opción de seguir introduciendo números
27  while salir == False:
28      numero = input("Introduce un número: ")
29      comprobarNumero(numero)
30      lista.append(int(numero))
31      opcion = input("¿Quieres introducir otro número?\nSI -> pulse una tecla\nNO -> pulse 'n'\n")
32      if opcion == 'n':
33          salir = True
34
35  #Al decidir no introducir más números, se muestra el valor máximo y el mínimo
36  calcularMaxMin(lista)
37
```

## Ejecución

```
Introduce un número: 24
¿Quieres introducir otro número?
SI -> pulse una tecla
NO -> pulse 'n'
s
Introduce un número: 13
¿Quieres introducir otro número?
SI -> pulse una tecla
NO -> pulse 'n'
s
Introduce un número: 98
¿Quieres introducir otro número?
SI -> pulse una tecla
NO -> pulse 'n'
s
Introduce un número: 65
¿Quieres introducir otro número?
SI -> pulse una tecla
NO -> pulse 'n'
n
El valor máximo es 98
El valor mínimo es 13
```

### Ejercicio 3

#### Código

```
Ejercicio03.py > ...
1  '''El día juliano correspondiente a una fecha es un número entero que indica los días que han
2  transcurrido desde el 1 de enero del año indicado. Queremos crear un programa principal que
3  al introducir una fecha nos diga el día juliano que corresponde. Para ello podemos hacer las
4  siguientes subrutinas:
5  - LeerFecha: Nos permite leer por teclado una fecha (día, mes y año).
6  - DiasDelMes: Recibe un mes y un año y nos dice los días de ese mes en ese año.
7  - EsBisiesto: Recibe un año y nos dice si es bisiesto.
8  - Calcular_Dia_Juliano: recibe una fecha y nos devuelve el día juliano.'''
9
10 #Método que comprueba que los datos introducidos sean numéricos
11 #En caso de no ser, se volverá a pedir que se introduzcan los datos
12 def comprobarDato(numero):
13     correcto = False
14     while correcto == False:
15         if numero.isdigit():
16             correcto = True
17         else:
18             print("ERROR -> Solo se permiten números enteros")
19             numero = input("Introduce un dato correcto: ")
20     return numero
21
22 #Método que comprueba si el año es bisiesto
23 def esBisiesto(year):
24     if ((year % 4 == 0 and not (year % 100 == 0)) or year % 400 == 0) == 0:
25         return False
26     else:
27         return True
28
```

```

29 |     ##Método que comprueba los días del mes
30 |     def diasDelMes(mes, year):
31 |         if mes == 1:
32 |             return 0
33 |         elif mes == 2:
34 |             return 31
35 |         elif mes == 3:
36 |             if esBisiesto(year):
37 |                 return 60
38 |             else:
39 |                 return 59
40 |         elif mes == 4:
41 |             if esBisiesto(year):
42 |                 return 91
43 |             else:
44 |                 return 90
45 |         elif mes == 5:
46 |             if esBisiesto(year):
47 |                 return 121
48 |             else:
49 |                 return 120
50 |         elif mes == 6:
51 |             if esBisiesto(year):
52 |                 return 152
53 |             else:
54 |                 return 151
55 |         elif mes == 7:
56 |             if esBisiesto(year):
57 |                 return 182
58 |             else:
59 |                 return 181
60 |         elif mes == 8:
61 |             if esBisiesto(year):
62 |                 return 213
63 |             else:
64 |                 return 212
65 |         elif mes == 9:
66 |             if esBisiesto(year):
67 |                 return 244
68 |             else:
69 |                 return 243
70 |         elif mes == 10:
71 |             if esBisiesto(year):
72 |                 return 274
73 |             else:
74 |                 return 273
75 |         elif mes == 11:
76 |             if esBisiesto(year):
77 |                 return 305
78 |             else:
79 |                 return 304
80 |         elif mes == 12:
81 |             if esBisiesto(year):
82 |                 return 335
83 |             else:
84 |                 return 334
85 |

```

```

#Método que muestra el día Juliano
def Calcular_Dia_Juliano(dia, mes, year):
    |     print("Para la fecha", str(dia), "/", str(mes), "/", str(year), ", el día juliano es", diasTotales + diaIntroducido)

#Se piden los datos al usuario
diaIntroducido = int(comprobarDato(input("Introduce un día: ")))
mesIntroducido = int(comprobarDato(input("Introduce un mes (en dígito): ")))
yearIntroducido = int(comprobarDato(input("Introduce un año: ")))
if yearIntroducido == 0:
    |     yearIntroducido = 1
#Se guardan los días que se llevan del año
diasTotales = int(diasDelMes(mesIntroducido, yearIntroducido))
#Se calcula el día Juliano
Calcular_Dia_Juliano(diaIntroducido, mesIntroducido, yearIntroducido)

```

## Ejecución

```
Introduce un día: 31
Introduce un mes (en dígito): 12
Introduce un año: 2001
Para la fecha 31 / 12 / 2001 , el día juliano es 365
```

```
Introduce un día: 3
Introduce un mes (en dígito): 6
Introduce un año: 2012
Para la fecha 3 / 6 / 2012 , el día juliano es 155
```

```
Introduce un día: 1
Introduce un mes (en dígito): 3
Introduce un año: 2001
Para la fecha 1 / 3 / 2001 , el día juliano es 60
```