

Boletín 3

Ejercicio 1

Código

```
Ejercicio01.py X
C: > Users > david > Desktop > DI- > Ejercicios03 > Ejercicio01.py
1  '''Realizar un programa que comprueba si una cadena leída por teclado comienza por una
2  subcadena introducida por teclado. '''
3
4  cadena = input("Introduce una cadena: ")
5  subcadena = input("Introduce una subcadena: ")
6
7  if (cadena.startswith(subcadena)):
8      print("La palabra empieza por la subcadena")
9  else:
10     print("La palabra no empieza por la subcadena")
```

Ejecución

```
Introduce una cadena: programación
Introduce una subcadena: pro
La palabra empieza por la subcadena
```

```
Introduce una cadena: programación
Introduce una subcadena: ción
La palabra no empieza por la subcadena
```

Ejercicio 2

Código

```
Ejercicio02.py X
C: > Users > david > Desktop > DI- > Ejercicios03 > Ejercicio02.py > ...
1  '''Pide una cadena y un carácter por teclado (valida que sea un carácter) y muestra cuantas veces
2  aparece el carácter en la cadena.'''
3
4  def comprobarCaracter(caracter):
5      estado = True
6      if (len(caracter) > 1):
7          estado = False
8      return estado
9
10 def comprobarCoincidencias():
11     contador = 0
12     for i in range(len(cadena)):
13         if (cadena[i] == caracter):
14             contador += 1
15     if(contador == 1):
16         print("El caracter " + caracter + " aparece 1 vez")
17     elif(contador > 1):
18         print("El caracter " + caracter + " aparece", contador, "veces")
19     else:
20         print("El caracter " + caracter + " no aparece en la cadena")
21
22 cadena = input("Introduce una cadena: ")
23 caracter = input("Introduce un caracter: ")
24 while (comprobarCaracter(caracter) == False):
25     print("Debe introducir un solo caracter")
26     caracter = input("Introduce un caracter: ")
27
28 comprobarCoincidencias()
```

Ejecución

```
Introduce una cadena: diseño de interfaces
Introduce un caracter: di
Debe introducir un solo caracter
Introduce un caracter: e
El caracter 'e' aparece 4 veces
```

Ejercicio 3

Código

```
Ejercicio03.py X
C: > Users > david > Desktop > DI- > Ejercicios03 > Ejercicio03.py > ...
1  '''Suponiendo que hemos introducido una cadena por teclado que representa una frase
2  (palabras separadas por espacios), realiza un programa que cuente cuantas palabras tiene.'''
3
4  def comprobarEspacios():
5      espacio = False
6      for i in range(len(cadena)):
7          if(cadena[i] == " "):
8              espacio = True
9      return espacio
10
11  cadena = input("Introduce una cadena: ")
12  while (comprobarEspacios() == False):
13      print("Debe introducir más de una palabra")
14      cadena = input("Introduce una cadena: ")
15  palabras = cadena.split()
16  print("La frase tiene", len(palabras), "palabras")
17
```

Ejecución

```
Introduce una cadena: buenos
Debe introducir más de una palabra
Introduce una cadena: buenos días por la mañana
La frase tiene 5 palabras
```

Ejercicio 04

Código

```
Ejercicio04.py X
C: > Users > david > Desktop > DI- > Ejercicios03 > Ejercicio04.py > ...
1  '''Realizar un programa que compruebe si una cadena contiene una subcadena. Las dos cadenas
2  se introducen por teclado'''
3
4  cadena = input("Introduce una cadena: ")
5  subcadena = input("Introduce una subcadena: ")
6
7  if (subcadena in cadena):
8      print("La palabra contiene la subcadena")
9  else:
10     print("La palabra no contiene la subcadena")
```

Ejecución

```
Introduce una cadena: diseño de interfaces
Introduce una subcadena: terf
La palabra contiene la subcadena
```

```
Introduce una cadena: diseño de interfaces
Introduce una subcadena: hinter
La palabra no contiene la subcadena
```