



Grado en Ingeniería Informática del Software Fundamentos de Informática

Solución a los ejercicios del tema 2.7: tipos estructurados (II)

Este documento incluye la solución a los ejercicios del documento Ejercicios del tema 2.7: tipos estructurados (II). Se recomienda que trates de hacer los ejercicios sin mirar a las soluciones primero, y que posteriormente compruebes tus respuestas. Ten en cuenta que un problema puede tener múltiples soluciones.

Ejercicio 1

Solución propuesta:

```
def capitalize(a):
    words = a.split(" ")
    for i in range(len(words)):
        word = words[i]
        words[i] = word[:1].upper() + word[1:]
    return " ".join(words)

a = input("Introduce la frase a capitalizar: ")
print(capitalize(a))
```

En la solución propuesta se hace uso de la función `.upper()`, debido a su versatilidad y a que es muy comúnmente utilizada en el manejo de cadenas de caracteres. No obstante, Python incluye otras funciones que podrían haber sido útiles, como por ejemplo `.capitalize()`, que convierte el primer carácter de una cadena en mayúscula, o la función `.title()`, que busca el mismo cometido que la función solicitada, solo que con algunas excepciones relacionadas con el uso de apóstrofes.

Ejercicio 2

Solución propuesta:

```
def to_leet(a):
    result = ""
    for char in a:
        if char in ["a", "á", "A", "Á"]:
            result = result + "4"
        elif char in ["e", "é", "E", "É"]:
            result = result + "3"
        elif char in ["i", "í", "I", "Í"]:
            result = result + "1"
        elif char in ["o", "ó", "O", "Ó"]:
            result = result + "0"
        else:
            result = result + char
    return result

a = input("Introduce la frase a convertir: ")
print(to_leet(a))
```



Ejercicio 3

Solución propuesta:

```
def to_bin(n):
    result = ""
    if n == 0:
        return "0"
    while n > 0:
        result = str(n % 2) + result
        n = n // 2
    return result

n = int(input("Introduce un valor entero no negativo: "))
print("La representación binaria de {} es {}".format(n, to_bin(n)))
```

Ejercicio 4

Solución propuesta:

```
def get_iban_cc(country, ccc):
    iban_base = ccc + country + "00"
    iban_base_num = ""
    for i in iban_base:
        if not i.isdigit():
            iban_base_num = iban_base_num + str(ord(i) - ord("A") + 10)
        else:
            iban_base_num = iban_base_num + i

    iban_cc = "{:02}".format(98 - int(iban_base_num) % 97)

    return iban_cc

def get_iban(country, ccc):
    iban_cc = get_iban_cc(country, ccc)
    iban = country + iban_cc + ccc

    iban_format = iban[:4]
    for i in range(4, len(iban), 4):
        iban_format = iban_format + " " + iban[i:(i + 4)]

    return iban_format

country = input("Introduce caracteres del país: ")
ccc = input("Introduce CCC: ")
print(get_iban(country, ccc))
```



Ejercicio 5

Solución propuesta:

```
def transpose(m):  
    result = []  
    for j in range(len(m[0])):  
        row = []  
        for i in range(len(m)):  
            row.append(m[i][j])  
        result.append(row)  
    return result  
  
m = [[0, 1, 2, 3], [4, 5, 6, 7], [8, 9, 10, 11]]  
print("La matriz traspuesta es {}".format(transpose(m)))  
print("La matriz original es {}".format(m))
```

Ejercicio 6

Solución propuesta:

```
def transpose(m):  
    result = []  
    for j in range(len(m[0])):  
        row = []  
        for i in range(len(m)):  
            row.append(m[i][j])  
        result.append(row)  
    m.clear()  
    m.extend(result)  
  
m = [[0, 1, 2, 3], [4, 5, 6, 7], [8, 9, 10, 11]]  
print("Antes de la función: {}".format(m))  
transpose(m)  
print("Después de la función: {}".format(m))
```



Ejercicio 7

Solución propuesta:

```
def identity(size):
    result = []
    for i in range(size):
        row = [0] * size
        row[i] = 1
        result.append(row)
    return result

size = int(input("Introduce el tamaño de la matriz identidad: "))

im = identity(size)
for i in range(len(im)):
    for j in range(len(im[i])):
        print(im[i][j], end=" ")
    print()
```

Ejercicio 8

Solución propuesta:

```
def symmetrical(m):
    for i in range(len(m)):
        for j in range(i + 1, len(m[i])):
            if m[i][j] != m[j][i]:
                return False
    return True

m = [[0, 1, 2], [1, 3, 4], [2, 4, 5]]
if symmetrical(m):
    print("La matriz {} es simétrica".format(m))
else:
    print("La matriz {} NO es simétrica".format(m))

m = [[0, 1, 2], [0, 1, 2], [0, 1, 2]]
if symmetrical(m):
    print("La matriz {} es simétrica".format(m))
else:
    print("La matriz {} NO es simétrica".format(m))
```