

# Tarea 5 - Estructuras de datos: Listas

## Curso de Python

### Ejercicio 1

Crea un programa que lea una secuencia de caracteres, guarde cada caracter en una posición de una lista y finalmente muestre la secuencia invertida.

#### Solución

```
s = input()
l = []

for c in s:
    l.append(c)

l.reverse()
print(l)
```

### Ejercicio 2

Crea un programa que lea dos strings de la misma longitud, los guarde intercalados en una lista. Por último, mostrar el contenido de la lista.

Por ejemplo, si introducimos `hola caracola` y `adios marieta`, debería mostrarse `haodleao sc amraarcioeltaa`.

#### Solución

```
s1 = input()
s2 = input()
l = []

if len(s1) == len(s2):
    for i in range(len(s1)):
        l.append(s1[i])
        l.append(s2[i])

    for item in l:
        print(item, end = "")

else:
    print("Los strings no tienen el mismo tamaño")
```

### Ejercicio 3

Crea un programa que lea un string y guarde en una lista todas las consonantes.

#### Solución

```
s = input()
l = []

for c in s.lower():
    if c == "a" or c == "e" or c == "i" or c == "o" or c == "u":
        continue
    l.append(c)

print(l)
```

### Ejercicio 4

Crea un programa que lea una palabra, la guarde en una lista y compruebe si se trata de un palíndromo.

#### Solución

```
word = input()
l = []
isPalindrom = True

for c in word:
    l.append(c)

n = len(l)
for i in range(int(n / 2)):
    if l[i] != l[n - (i + 1)]:
        isPalindrom = False

print("¿Es {} palíndromo? {}".format(word, "Sí" if isPalindrom else "No"))
```

### Ejercicio 5

Crea un programa que lea una matriz 3 x 3 y devuelva el máximo de cada fila.

#### Solución

```
A = []
for i in range(3):
    A.append([])
    for j in range(3):
```

```

        A[i].append(float(input("Introduce el elemento ({}, {}) ".format(i, j))))

for i in range(3):
    A[i].sort(reverse = True)
    print("El máximo de la fila {} es {}".format(i, A[i][0]))

```

## Ejercicio 6

Crea un programa que lea un entero,  $n$ , de teclado y construya una matriz de tamaño  $n \times n$ . Cada posición debe contener su orden en la matriz (desde 0 hasta  $n^2 - 1$ ). Por ejemplo, si  $n$  es 3, deberá crearse la matriz

```

n = int(input())
m = []

for i in range(n):
    m.append([])
    for j in range(n):
        m[i].append(n * i + j)

for i in range(n):
    print(m[i])

```

## Ejercicio 7

Crea una matriz de  $n \times m$  y asigna los valores manualmente. El programa debe indicar si la suma de cada columna es el mismo valor.

### Solución

```

n = int(input())
m = int(input())

A = []
for i in range(n):
    A.append([])
    for j in range(m):
        A[i].append(int(input("Elemento ({}, {}) ".format(i, j))))

sums = []
for j in range(m):
    sum = 0
    for i in range(n):
        sum += A[i][j]
    sums.append(sum)

sameSum = True
for i in range(m - 1):
    if sums[i] != sums[i+1]:
        sameSum = False

```

```
print("¿La suma de las columnas es igual? {}".format("Sí" if sameSum else "No"))
```

## Ejercicio 8

Crear un programa determina si la matriz introducida manualmente (tanto sus dimensiones como los elementos) se trata de la matriz identidad. Recuerda que la matriz identidad debe ser una matriz cuadrada.

### Solución

```
n = int(input("Número de filas: "))
m = int(input("Número de columnas: "))

A = []
for i in range(n):
    A.append([])
    for j in range(m):
        A[i].append(int(input("Elemento ({}, {}) ".format(i, j))))

isIdentity = True
if n == m:
    for i in range(n):
        for j in range(n):
            if i == j and A[i][j] != 1:
                isIdentity = False
            if i != j and A[i][j] != 0:
                isIdentity = False
else:
    isIdentity = False
print("La matriz no es cuadrada. No puede ser la matriz identidad.")

print("¿La matriz introducida es la matriz identidad? {}".format("Sí" if isIdentity else "No"))
```

## Ejercicio 9

Dada una matriz formada con listas, realiza un programa que calcule la matriz transpuesta.

### Solución

```
A = [[1, 2, 3], [4, 5, 6], [7, 8, 9]]

T = []
for i in range(len(A[0])):
    T.append([])
    for j in range(len(A)):
        T[i].append(A[j][i])
```

```
for i in range(len(T)):
    for j in range(len(T[0])):
        print(T[i][j], end = " ")
    print("")
```

## Ejercicio 10

Crea un programa que pida al usuario la dimensión y cree la matriz identidad de orden correspondiente con numpy.

### Solución

```
import numpy as np
n = int(input())
I = np.zeros((n, n))
for i in range(n):
    I[i, i] = 1

print(I)
```