

Escriba un método generar(n) que genere un arreglo de tamaño n con todos sus elementos aleatorios entre 1 y 100.

```
In [ ]: import random

def generar(n):
    list = random.sample(range(1, 100), n)
    print(list)

generar(10)
```

```
[40, 80, 38, 7, 84, 88, 92, 75, 90, 78]
```

Escriba un método mostrar(int[] t) que muestra todos los elementos de un arreglo t.

```
In [ ]: def mostrar(arreglo):
    print(arreglo)
    for item in arreglo:
        print(item)

arr = ["manzana", "naranja", "sandia", "piña", "cereza"]
mostrar(arr)
```

```
['manzana', 'naranja', 'sandia', 'piña', 'cereza']
manzana
naranja
sandia
piña
cereza
```

Escriba un método reversa(int[] t) que invierte el orden de todos los elementos de un arreglo t (sin crear arreglos auxiliares)

```
In [ ]: def reversa(arreglo):
    arreglo.reverse()
    for item in arreglo:
        print(item)

arr = ["manzana", "naranja", "sandia", "piña", "cereza"]
reversa(arr)
```

```
cereza
piña
sandia
naranja
manzana
```

Escriba un método int minArreglo(int[] t) que devuelve el elemento más pequeño de un arreglo.

```
In [ ]: # Opcion 1
def minArreglo(arreglo):
```

```

    arreglo.sort()
    return arreglo[0]

# Opcion 2
def minArreglo2(arreglo):
    return min(arreglo)

arr = ["manzana", "naranja", "sandia", "piña", "cereza"]
#arr = [7, 3, 5, 1, 4]

minArreglo(arr)

```

Out[]: 'cereza'

Escriba un método `int mediaArreglo(int[] t)` que devuelve la media aritmética de los elementos de un arreglo.

```

In [ ]: def mediaArreglo(arreglo):
        return sum(arreglo)/len(arreglo)

arr = [7, 3, 5, 1, 4]

mediaArreglo(arr)

```

Out[]: 4.0

Escriba un método `int[] ocurrencias(int[] t)` que devuelve un nuevo arreglo `s` tal que `s[i]` es el número de veces que el entero `i` aparece en `t`

```

In [ ]: nuevo_arreglo = []
def ocurrencias(arreglo):
    for i in arreglo:
        nveces = arreglo.count(i)
        nuevo_arreglo.append(nveces)
    print("Arr. original: ", arreglo)
    print("NºOcurrencias: ", nuevo_arreglo)

arreglo = [1, 2, 3, 1, 2]
ocurrencias(arreglo)

```

```

Arr. original:  [1, 2, 3, 1, 2]
NºOcurrencias:  [2, 2, 1, 2, 2]

```