

Ejercicio de Java 002: Ejercicios de Calentamiento

Roshka

Bootcamp

Tiempo Estimado del Ejercicio: 4 horas

Ejercicios

1. Cargar un array de manera aleatoria con 10 números enteros del -5 al 5. Imprimirlo en pantalla y computar cuál es el mayor elemento del vector.

```
import Foundation

func generarArrayAleatorioYEncontrarMayor() {

    var numeros: [Int] = []

    var mayorElemento: Int?

    for _ in 1...10 {

        let numeroAleatorio = Int.random(in: -5...5)

        numeros.append(numeroAleatorio)

    }

    print("Array de números aleatorios: \(numeros)")

    if let maxElemento = numeros.max() {

        mayorElemento = maxElemento

        print("El mayor elemento del array es: \(mayorElemento!)")    }

    }

    generarArrayAleatorioYEncontrarMayor()
```

2. Cargar un array de manera aleatoria con 100 números enteros del -30 al 30. Imprimirlo en pantalla y computar cuál es el elemento que más veces se repite, y cuáles son los números que no están presentes (si es que hay alguno).

```
import Foundation

func generarArrayAleatorioYAnalizar() {

    var numeros: [Int] = []

    for _ in 1...100 {

        let numeroAleatorio = Int.random(in: -30...30)
```

```

        numeros.append(numeroAleatorio)
    }

    print("Array de números aleatorios: \${numeros}")

    var conteo: [Int: Int] = [:]

    for numero in numeros {

        conteo[numero, default: 0] += 1

    }

    let maxElemento = conteo.max(by: { $0.value < $1.value })

    if let (numeroMasFrecuente, cantidad) = maxElemento {

        print("El número que más se repite es \${numeroMasFrecuente} con \${cantidad} ocurrencias.")

    }

    let todosLosNumeros = Set(-30...30)

    let numerosPresentes = Set(numeros)

    let numerosFaltantes = todosLosNumeros.subtracting(numerosPresentes)

    if numerosFaltantes.isEmpty {

        print("Todos los números del rango [-30, 30] están presentes.")

    } else {

        print("Los números que no están presentes son: \${numerosFaltantes.sorted()}")

    }

}

generarArrayAleatorioYAnalizar()

```

3. Hacer una función que, dada una palabra (String) o frase, diga si la misma es palíndromo o no. Una palabra/frase palíndromo es aquella que se lee igual tanto de atrás para adelante, como de adelante para atrás. Ejemplos de palíndromos: "MADAM", "RACECAR", "AMORE, ROMA", "BORROW OR ROB", "WAS IT A CAR OR A CAT I SAW?".

```
import Foundation
```

```
func esPalindromo(_ cadena: String) -> Bool {
```

```
    let caracteresFiltrados = cadena.lowercased().filter { $0.isLetter }

    return caracteresFiltrados == String(caracteresFiltrados.reversed())
}

let frasesParaProbar = [

    "MADAM",

    "RACECAR",

    "AMORE, ROMA",

    "BORROW OR ROB",

    "WAS IT A CAR OR A CAT I SAW?"

]

for frase in frasesParaProbar {

    if esPalindromo(frase) {

        print("\(frase)" es un palíndromo.")

    } else {

        print("\(frase)" no es un palíndromo.")

    }

}
```

4. Dada una cadena de caracteres (String) de longitud desconocida que tiene solamente dígitos, crear un array de N elementos (donde N es el tamaño de la cadena) que tenga cada uno de los valores numéricos de los dígitos.

import Foundation

```
func convertirCadenaADigitos(_ cadena: String) -> [Int] {
```

```
    var numeros: [Int] = []
```

```
    for caracter in cadena {
```

```
        if let digito = caracter.wholeNumberValue {
```

```
            numeros.append(digito)
```

```
        }
```

```
    }
```

```
    return numeros
```

```
}
```

```
let cadenaDeDigitos = "1234567890"
```

```
let arrayDeNumeros = convertirCadenaADigitos(cadenaDeDigitos)
```

```
print("Array de números: \(arrayDeNumeros)")
```