

Ejercicio 2: Kotlin Core

1. Crea un programa que te pida una cantidad en miligramos para una poción multijugos, el valor debe ser de tipo entero, si el valor es mayor a 100 imprime “¡Felicitaciones, es una buena poción multijugos!”, de lo contrario imprime “La poción es mediocre, sangre sucia inmunda”.
2. Escribe un programa que te diga si un carro de Uber puede iniciar su recorrido, para esto se necesitan dos cosas, que el conductor esté cerca y que esté disponible, el programa te pedirá dos valores, la distancia del conductor en kilómetros y su disponibilidad, donde false = no disponible y true = disponible, según los datos que insertes imprime lo siguiente:
 - Si la distancia es menor o igual a 0.5 km y el conductor está disponible, imprime “Listo para iniciar recorrido”
 - Si la distancia es menor o igual a 0.5 km y el conductor NO está disponible, imprime, “Conductor cercano, pero no disponible”
 - Si la distancia es mayor a 0.5 km y el conductor está disponible, imprime, “Conductor disponible pero muy lejos, aplicarán tarifas más altas”
 - Si la distancia es mayor a 0.5 km y el conductor NO está disponible, imprime, “No hay conductores disponibles”
 - Usa if-else para esto
 - Para negar una validación debes usar el símbolo !, por ejemplo if (!isAvailable)
 - No olvides imprimir las instrucciones antes de pedir las variables de entrada.
RETO OPCIONAL: Si no se cumplen las condiciones de “Listo para iniciar recorrido” vuelve a pedir los datos de distancia y disponibilidad.
3. Haz un ciclo for y while que obtenga la sumatoria de los números hasta n, ejemplo, para 5 debes obtener 15 (1+2+3+4+5), para 3 debes obtener 6. Imprime el resultado así: “La suma es 15” usando formatos de String
4. Escribe un programa que imprima “¿Cómo es el clima de hoy?”, dependiendo del número que insertes te imprima los siguientes valores
 - 1 = “Soleado”
 - 2 = “Nublado”
 - 3 = “Lluvioso”
 - 4 = “Tormentoso”
 - 5 = “Nevado”

Si insertas cualquier otro valor debe imprimir “Pregúntale a Google”
Recomendación: Usa When.
5. Para un Arreglo de String de títulos de películas, imprime el título de la película con el título más largo. Por ejemplo si tenemos un arreglo con los valores: {“Jumanji”, “Toy Story”, “Pulp Fiction”, “Batman: El caballero de la noche”, “Kill Bill”} Debería imprimir “Batman: El caballero de la noche”.

RETO OPCIONAL: Imprime el índice del título más largo, para el ejemplo anterior, debería imprimir el 3, porque es el índice de “Batman: El caballero de la noche”

6. **EJERCICIO RETO SUPER SAYAYÍN NIVEL ULTRA INSTINCT:** Escribe un programa que para un número N, imprima los números pares desde 1 hasta N, por ejemplo si N = 6, debe imprimir “2, 4, 6”. Hazlo con un for o un while. Si el número es menor o igual a 0, debes imprimir “Inserta un número positivo”
TIP: Para saber si un número es par, debes obtener el residuo de ese número al ser dividido entre 2, si el residuo es 0, entonces el número es par. Por ejemplo si quieres saber si 4 es par, divides $4/2 = 2$ y te sobran 0, entonces 4 es par, en cambio para $5/2 = 2$ y sobra 1, entonces es impar, para obtener el residuo de una división en Java puedes utilizar %, ejemplo: $4 \% 2 = 0$ y $5 \% 2 = 1$.

Respuestas

1.

```
▶ 1 fun main() {
    2     println("Inserta mg de antimonio para una poción multijugos")
    3     val scanner = Scanner(System.`in`)
    4     val mg = scanner.nextInt()
    5     if (mg > 100) {
    6         println("¡Felicidades, es una buena poción multijugos!")
    7     } else {
    8         println("La poción es mediocre, sangre sucia inmunda.")
    9     }
}
```

2.

```
3 ▶ 1 fun main() {
4     2     val scanner = Scanner(System.`in`)
5     3     println("Inserta la distancia al automóvil")
6     4     val distanceToCarInKm = scanner.nextDouble()
7     5     println("¿El conductor está disponible? (Escribe true o false)")
8     6     val isAvailable = scanner.nextBoolean()
9
10    7     if (distanceToCarInKm <= 0.5 && isAvailable) {
11        8         println("Listo para iniciar recorrido")
12    } else if (distanceToCarInKm <= 0.5 && !isAvailable) {
13        9         println("Conductor cercano, pero no disponible")
14    } else if (distanceToCarInKm > 0.5 && isAvailable) {
15        10        println("Conductor disponible pero muy lejos, aplicarán tarifas más altas")
16    } else {
17        11        println("No hay conductores disponibles")
18    }
19 }
```

CON RETO OPCIONAL

```
▶ | fun main() {
|   val scanner = Scanner(System.`in`)

|   do {
|       println("Inserta la distancia al automóvil")
|       val distanceToCarInKm = scanner.nextDouble()
|       println("¿El conductor está disponible? (Escribe true o false)")
|       val isAvailable = scanner.nextBoolean()

|       if (distanceToCarInKm <= 0.5 && isAvailable) {
|           println("Listo para iniciar recorrido")
|       } else if (distanceToCarInKm <= 0.5 && !isAvailable) {
|           println("Conductor cercano, pero no disponible")
|       } else if (distanceToCarInKm > 0.5 && isAvailable) {
|           println("Conductor disponible pero muy lejos, aplicarán tarifas más altas")
|       } else {
|           println("No hay conductores disponibles")
|       }
|   } while(distanceToCarInKm > 0.5 || !isAvailable)
| }
```

3.

```
1 ▶ | fun main() {
|   val n = 5
|   var total = 0
|   for (i in 0..n) {
|       total += i
|   }
|
|   println("Total is $total")
| }
```

4.

```
▶ fun main() {
    val scanner = Scanner(System.`in`)
    println("¿Cómo es el clima de hoy?")
    val weather = scanner.nextInt()

    when(weather) {
        1 -> println("El clima es soleado")
        2 -> println("El clima es nublado")
        3 -> println("El clima es lluvioso")
        4 -> println("El clima es tormentoso")
        5 -> println("El clima es nevado")
        else -> println("Pregúntale a Google")
    }
}
```

5.

Solución A

```
▶ fun main() {
    val titleArray = arrayOf("Jumanji", "Toy Story", "Pulp Fiction", "Batman: El caballero de la noche", "Kill Bill")

    var maxLength = 0
    var longestTitle = ""
    for (title in titleArray) {
        val titleLength = title.length
        if (titleLength > maxLength) {
            maxLength = titleLength
            longestTitle = title
        }
    }

    println(longestTitle)
}
```

Solución B (Magia de Kotlin, lo veremos más adelante)

```
3 ▶ fun main() {
4     val titleArray = arrayOf("Jumanji", "Toy Story", "Pulp Fiction", "Batman: El caballero de la noche", "Kill Bill")
5
6     val longestTitle = titleArray.maxBy { it.length }
7
8     println(longestTitle)
9
10 }
```

5. RETO:

```
▶ fun main() {
    val titleArray = arrayOf("Jumanji", "Toy Story", "Pulp Fiction", "Batman: El caballero de la noche", "Kill Bill")

    var maxLength = 0
    var longestTitle = ""
    var longestTitleIndex = 0
    for (index in titleArray.indices) {
        val title = titleArray[index]
        val titleLength = title.length
        if (titleLength > maxLength) {
            maxLength = titleLength
            longestTitle = title
            longestTitleIndex = index
        }
    }

    println("Longest title is $longestTitle and index is $longestTitleIndex")
}
```

6.

```
3 ▶ fun main() {
4     val scanner = Scanner(System.`in`)
5     println("Inserta un número entero mayor a 0 para obtener los números hasta ese número")
6     val n = scanner.nextInt()

7     if (n <= 0) {
8         println("Debes insertar un número positivo")
9     } else {
10        for (i in 1..n) {
11            if (i % 2 == 0) {
12                println(i)
13            }
14        }
15    }
16 }
```