Ejercicio 1:

Sumar los números del 1 al 100 usando un bucle `for`

Escribe una función llamada `sumarNumeros` que utilice un bucle `for` para sumar todos los números del 1 al 100. Luego, imprime el resultado.

Pista: Usa un bucle `for` con un rango de 1 a 100.

Respuesta Alumno

```
// Ejercicio 1. (Sumar los números del 1 al 100 usando un bucle for)
  var sum = 0;

  for(i in 1..100) {
     sum += i;
  }

  println("La suma de los números del 1 al 100 es $sum")
```

Ejercicio 2:

Contar números pares entre 1 y 50 con un bucle `while`

Crea una función llamada `contarNumerosPares` que utilice un bucle `while` para contar cuántos números pares hay entre 1 y 50. Imprime el resultado.

Pista: Incrementa un contador solo cuando el número sea par.

Respuesta Alumno

```
// Ejercicio 2. (Sumar la cantidad de números pares que hay entre 1 y 50).
    fun contarNumerosPares() : Int {
       var aux = 1
        var contador = 0

        while(aux <= 50) {
            if(aux % 2 == 0) {
                contador++;
            }
            aux++
        }

        return contador
    }
```

Ejercicio 3:

Mostrar un menú hasta que el usuario seleccione salir (bucle `do-while`)

Crea una función llamada `mostrarMenu` que imprima un menú simple con tres opciones:

1. Sumar dos números.
2. Restar dos números.
3. Salir.

El menú debe repetirse hasta que el usuario elija la opción "Salir". Utiliza un bucle `do-while` para esto.

Pista: Usa la clase `Scanner` para leer las entradas del usuario.

Respuesta Alumno

```
// Ejercicio 3. (Mostrar un menú hasta que el usuario decida salir)
 fun mostrarMenu() {
    val scanner = Scanner(System.`in`)
    do {
       println("------------------")
       println(" Menú interactivo")
       println("------------------")
       println("1. Sumar dos números.")
       println("2. Restar dos números.")
       println("0. Salir.")
       var opcion = scanner.nextInt()
    } while(opcion != 0)
 }
```

Ejercicio 4:

Imprimir una tabla de multiplicar usando un bucle for anidado

Escribe una función llamada `imprimirTablaMultiplicar` que imprima la tabla de multiplicar del 1 al 10 utilizando bucles `for` anidados.

Pista: Un bucle controla el número base, y el otro controla los multiplicadores.

Respuesta Alumno

```
//Ejercicio 4. (Imprimir tabla de multiplicar)
  fun imprimirTablaMultiplicar() {

     for(i in 1..10) {
       for (j in 1..10) {
          var operacion = i * j
          println("$i * $j = $operacion")
       }
         println("\n")
     }
  }
```

Ejercicio 5:

Invertir una cadena con un bucle `for`

Escribe una función llamada `invertirCadena` que reciba una cadena y la invierta utilizando un bucle `for`.

Pista: Recorre la cadena desde el final hasta el principio.

Respuesta Alumno

```
// Ejercicio 5. (Invertir una cadena)
  fun invertirCadena (cadena : String) : String {
     var cadenaInvertida = ""
     for(i in (cadena.length - 1) downTo 0 ) {
        cadenaInvertida += cadena[i]
     }

     return cadenaInvertida
  }
```

Ejercicio 6:

Encontrar el primer número múltiplo de 7 mayor que 100 con un bucle `while`

Crea una función llamada `primerMultiploDeSiete` que encuentre el primer número múltiplo de 7 mayor que 100 utilizando un bucle `while`. Imprime el resultado.

Respuesta Alumno

```
// Ejercicio 6. (Encontrar primer múltiplo de 7 mayor que 100)
 fun primerMultiploDeSiete() : Int {
    var aux = 1
    var encontrado = false

    while(!encontrado) {
       if(aux % 7 == 0 && aux > 100) {
          encontrado = true
          return aux
       }
       aux++
    }

    return 0
 }
```

Ejercicio 7:

Controlar un bucle con `break` y `continue`

Crea una función llamada `bucleConControl` que recorra los números del 1 al 10:

- Si el número es divisible por 3, usa `continue` para saltarte esa iteración.
- Si el número es mayor que 8, usa `break` para salir del bucle.

Imprime los números que no se salten ni rompan el bucle.

Respuesta Alumno

```
// Ejercicio 7. (Controlar un bucle con continue y break)
  fun bucleConControl() {
     for(i in 1..10) {
        if (i % 3 == 0) {
           continue
        }
        if (i > 8) {
           break
        }

        println(i)
     }
  }
```