

1.- ¿De qué tipo son cada una de estas expresiones?

`val rating = 9.7`

`val bugsPerLineOfCode = 0.3f`

`val moleculeCount = 13L`

`const val QUIT_SYMBOL = 'q'`

`val ratio = 17 / 5`

`val offset = "Hello, World".length`

`val punctuation = listOf("comma", "period", "semicolon", "dash")`

```
val status = if(finishedWork()) {  
    goHome()  
    "Going home..."  
} else {  
    writeCode()  
    "Writing perfectly clean code..."  
}
```

`val grades = 1..6`

`val lowerCaseLatin = 'a'..'z'`

2.- Considera el siguiente código y corrígelo para que compile

```
edad = 16  
print(edad)  
edad = 30  
print(edad)
```

3.- ¿Qué problema hay en este código?

```
val nombre = "María"  
nombre += " Arrastia"
```

4.- ¿Cuales de las siguientes declaraciones son válidas?

```
val character: Character = "Perro"  
val character: Char = 'd'  
val string: String = "Perro"  
val string: String = 'd'
```

5.- ¿Cual es el valor de las siguientes variables?

```
val respuesta1 = true && true  
val respuesta2 = false || false  
val respuesta3 = (true && 1 != 2) || (4 > 3 && 100 < 1)  
val respuesta4 = ((10 / 2) > 3) && ((10 % 2) == 0)
```

6.- Crea una función que convierta una temperatura expresada en °C a otra expresada e °F. y otra función que haga lo contrario

7.- Implementa esta función que elimine la primera ocurrencia del item en la lista y devuelva la lista resultante

```
fun eliminaUno(item: Int, lista: List<Int>): List<Int>
```

8.- Implementa esta función que elimine todas las ocurrencias de item en la lista

```
fun elimina(item: Int, lista: List<Int>): List<Int>
```

9.- Transforma si puedes los bucles for en bucles while y viceversa.

```
for (word in sentence.split(" ")) {  
    print("$word. ")  
}
```

```
for (gradeBoundary in 55..100 step 5) {  
    println("Next grade: ${gradeBoundary - 5} - ${gradeBoundary}")  
}
```

```
while (userInput != ":q") {  
    print("> ")  
    val userInput = readLine() ?: ""  
}
```

10.- En el siguiente bucle for, ¿Que valor tendrá suma y cuantas iteraciones se darán?

```
var suma = 0
for(i in 0..5) {
    suma += i
}
```

11.- Muestra por pantalla: 0.0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0.

12.- Dado un Map que relaciona nombres de personas con una lista de los nombres de sus hijos:

```
val parentsToChildren = mapOf(
    "Susan" to listOf("Kevin", "Katie"),
    "Marcus" to listOf("Claire"),
    "Kate" to emptyList(),
    "Mike" to listOf("Jake", "Helen", "John")
)
```

Selecciona todas las personas cuyo nombre comience por M y que tengan al menos un hijo.

13.- ¿Cuales de las siguientes sentencias son válidas?

```
var name: String? = "Ray"
var age: Int = null
val distance: Float = 26.7
var middleName: String? = null
```

14.- Escribe una función que calcule el máximo y el mínimo de una lista de enteros. Calcula tu mismo los valores sin usar las funciones min o max. Devuelve ambos valores en un objeto de tipo Pair

**15.- Crea un proyecto usando programación orientada a objetos para modelar una universidad. Deberás implementar lo siguiente:
Universidad, Profesor, Estudiante y Asignatura
Estos deberán cumplir lo siguiente:**

- La universidad tendrá un nombre y un año de fundación. Podrá contratar profesores e inscribir a nuevos estudiantes añadiéndolos a una lista de estudiantes o profesores.
- El profesor tendrá un nombre, edad y salario anual. Podrá enseñar asignaturas y examinar estudiantes lo que le dará al estudiante una nota para ese curso.
- El estudiante tendrá un nombre, una edad y un identificador, una nota por cada curso de los que haya hecho un examen y una probabilidad de aprobar que empezará en el 50%. El estudiante podrá matricularse en asignaturas y examinarse de ellas. Podrá estudiar lo que aumentará su probabilidad de aprobar un 2%. También podrá ir de fiesta y eso hará que baje la probabilidad de aprobar un 2%.
- Cada asignatura tiene un título y una descripción.

16.- Reescribe esta clase para que:

```
class Helado {  
    val nombre: String  
    val ingredientes: ArrayList<String>  
}
```

- 1. La propiedad nombre tenga un valor por defecto**
- 2. Implementa una inicialización perezosa de la lista de ingredientes.**

17.- Crea tres clases llamadas A, B y C. Haz que C herede de B y B herede de A. Haz que cada clase tenga un bloque init que muestre por pantalla: “Soy X” siendo X el nombre de la clase. Crea una instancia de la clase C y ejecuta. ¿En que orden se ejecutan los inicializadores?