Gerardo Daniel Naranjo Gallegos, A01209499
Desarrollo de Aplicaciones Móviles
24/02/2020

Tarea 01: Introducción a Kotlin

Contenido

Objetivo

El Objetivo de esta asignación es que el estudiante se enfrente a las bases del lenguaje de programación Kotlin resolviendo problemas en los cuales se sienta seguro y que le permitan entender el funcionamiento de este.

Instrucciones

De la página: Ninety Nine Kotlin Problems, seleccionar 2 problemas 1 fácil y uno medio o difícil. Problemas fáciles: P05 - P28; problemas medios: P30 - P50; problemas difíciles: P55 – 99.

Resolver los problemas dentro de un archivo con extensión ".kts" (nota: Verificar que el archivo no sea ".kt"); ".kts" se refiere a Kotlin Script, mientras que ".kt" se refiere a solo un archivo Kotlin.

Evaluación

Se evaluará que el archivo contenga:

- Nombre y matrícula dentro del archivo.
- Números de problemas a resolver.
- Pseudocódigo o algoritmo de los problemas a resolver.
- Funcionalidad de los problemas a resolver.
- Casos de prueba:
    - Todos los dados por el problema.
    - Al menos 2 realizados por el estudiante.

Rúbrica

La siguiente rúbrica será utilizada para cada uno de los problemas:

Concepto	Porcentaje
Pseudocódigo	30%
Código	40%
Funcionamiento	30%

Fecha de entrega

27/02/2020 11:59 a través de GitHub Classroom.

Problemas asignados a resolver

Los problemas por resolver son el 14 y el 39.

Respuesta del problema 14

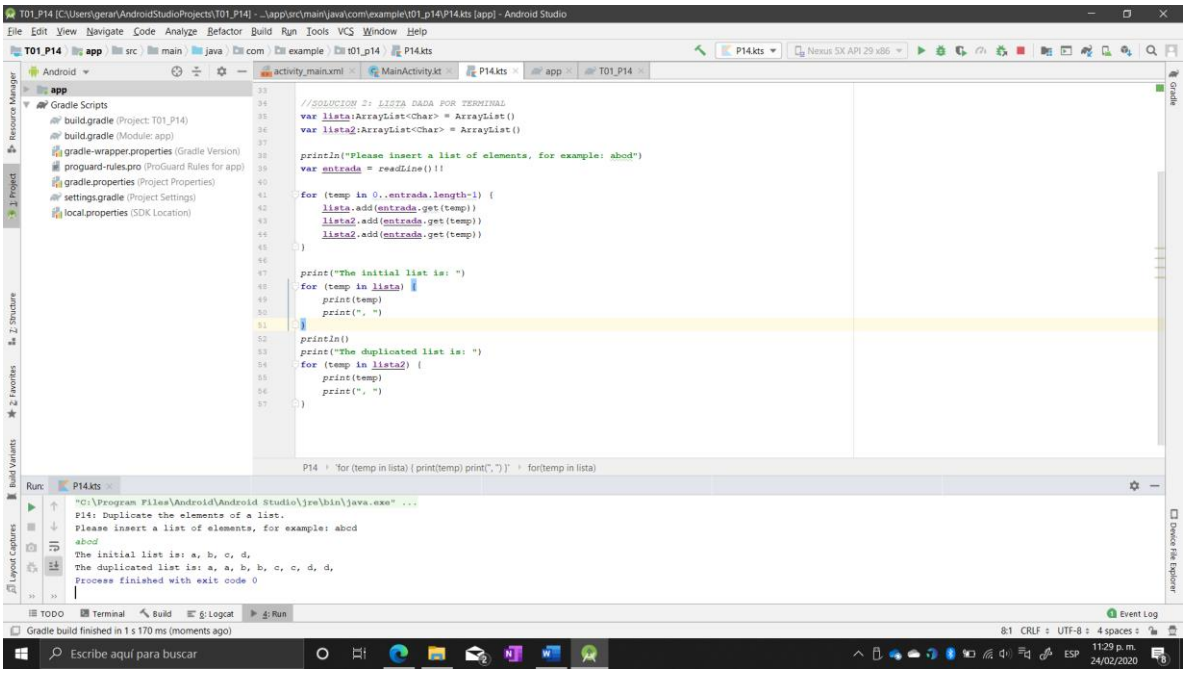
El **pseudocódigo** utilizado para este problema es:

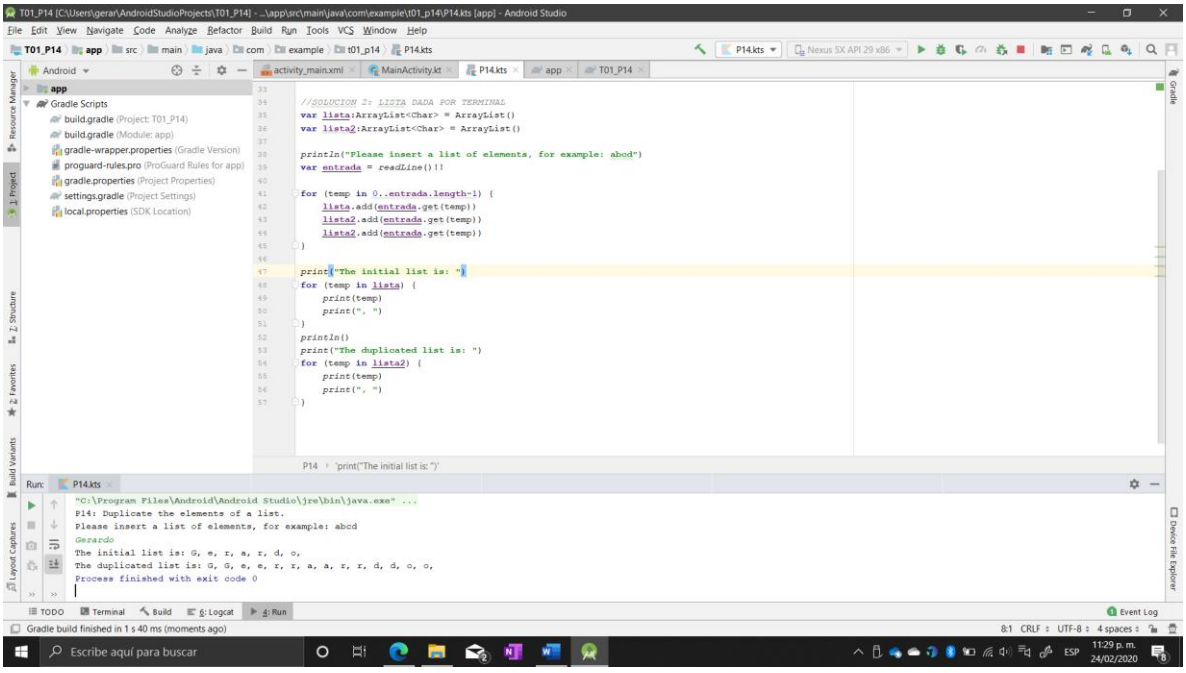
1) Crear la variable de la lista de entrada.
2) Crear la variable de la lista de salida.
3) Crear una variable para almacenar la entrada de datos (desde consola).
4) Obtener elementos de la lista de consola.
5) Recorrer en un ciclo *for* los datos ingresados desde consola y agregarlos 1 vez a la lista de entrada y 2 veces a la lista de salida.
6) Imprimir ambas listas.

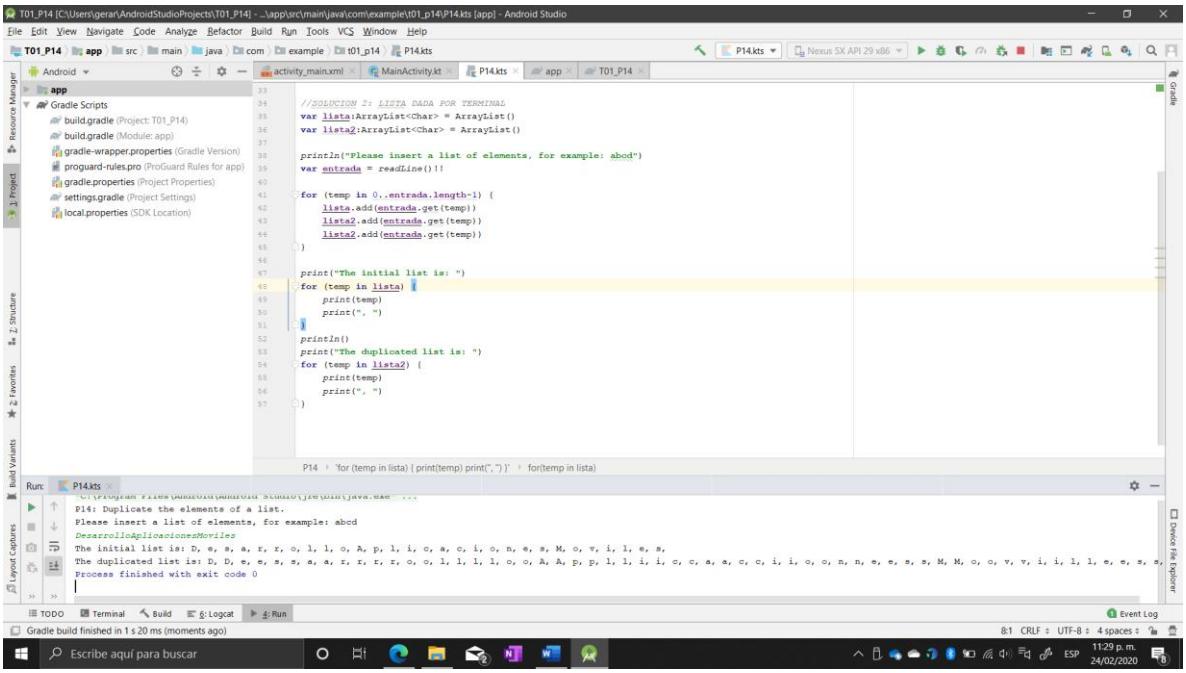
El **código** utilizado es el siguiente:

```
package com.example.t01_p14
println("P14: Duplicate the elements of a list.")
var lista:ArrayList<Char> = ArrayList()
var lista2:ArrayList<Char> = ArrayList()
println("Please insert a list of elements, for example: abcd")
var entrada = readLine()!!
for (temp in 0..entrada.length-1) {
    lista.add(entrada.get(temp))
    lista2.add(entrada.get(temp))
    lista2.add(entrada.get(temp))
}
print("The initial list is: ")
for (temp in lista) {
    print(temp)
    print(", ")
}
println()
print("The duplicated list is: ")
for (temp in lista2) {
    print(temp)
    print(", ")
}
```

Los **casos de prueba** dados son el asignado por el problema “a,b,c,d” y los dados por el alumno “Gerardo” y “DesarrolloAplicacionesMoviles”, como se pueden observar en las siguientes capturas de pantalla, a manera de evidencia:


T01_P14 [C:\Users\gerar\AndroidStudioProjects\T01_P14] - ...\app\src\main\java\com\example\t01_p14\P14.kts [app] - Android Studio
//SOLUCION 2: LISTA DADA POR TERMINAL
var lista:ArrayList<Char> = ArrayList()
var lista2:ArrayList<Char> = ArrayList()
println("Please insert a list of elements, for example: abcd")
var entrada = readLine()!!
for (temp in 0..entrada.length-1) {
lista.add(entrada.get(temp))
lista2.add(entrada.get(temp))
lista2.add(entrada.get(temp))
}
print("The initial list is: ")
for (temp in lista) {
print(temp)
print(", ")
}
println()
print("The duplicated list is: ")
for (temp in lista2) {
print(temp)
print(", ")
}
"C:\Program Files\Android\Android Studio\jre\bin\java.exe" ...
P14: Duplicate the elements of a list.
Please insert a list of elements, for example: abcd
abcd
The initial list is: a, b, c, d,
The duplicated list is: a, a, b, b, c, c, d, d,
Process finished with exit code 0
Gradle build finished in 1 s 170 ms (moments ago)


T01_P14 [C:\Users\gerar\AndroidStudioProjects\T01_P14] - ...\app\src\main\java\com\example\t01_p14\P14.kts [app] - Android Studio
//SOLUCION 2: LISTA DADA POR TERMINAL
var lista:ArrayList<Char> = ArrayList()
var lista2:ArrayList<Char> = ArrayList()
println("Please insert a list of elements, for example: abcd")
var entrada = readLine()!!
for (temp in 0..entrada.length-1) {
lista.add(entrada.get(temp))
lista2.add(entrada.get(temp))
lista2.add(entrada.get(temp))
}
print("The initial list is: ")
for (temp in lista) {
print(temp)
print(", ")
}
println()
print("The duplicated list is: ")
for (temp in lista2) {
print(temp)
print(", ")
}
"C:\Program Files\Android\Android Studio\jre\bin\java.exe" ...
P14: Duplicate the elements of a list.
Please insert a list of elements, for example: abcd
Gerardo
The initial list is: G, e, r, a, r, d, o,
The duplicated list is: G, G, e, e, r, r, a, a, r, r, d, d, o, o,
Process finished with exit code 0
Gradle build finished in 1 s 40 ms (moments ago)


T01_P14 [C:\Users\gerar\AndroidStudioProjects\T01_P14] - ...\app\src\main\java\com\example\t01_p14\P14.kts [app] - Android Studio
File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
T01_P14 app src main java com example t01_p14 P14.kts
activity_main.xml MainActivity.kt P14.kts app T01_P14
Android
app
Gradle Scripts
build.gradle (Project: T01_P14)
build.gradle (Module: app)
gradle-wrapper.properties (Gradle Version)
proguard-rules.pro (ProGuard Rules for app)
gradle.properties (Project Properties)
settings.gradle (Project Settings)
local.properties (SDK Location)
//SOLUCION 2: LISTA DADA POR TERMINAL
var lista:ArrayList<Char> = ArrayList()
var lista2:ArrayList<Char> = ArrayList()
println("Please insert a list of elements, for example: abcd")
var entrada = readLine()!!
for (temp in 0..entrada.length-1) {
lista.add(entrada.get(temp))
lista2.add(entrada.get(temp))
}
print("The initial list is: ")
for (temp in lista) {
print(temp)
print(", ")
}
println()
print("The duplicated list is: ")
for (temp in lista2) {
print(temp)
print(", ")
}
P14 for (temp in lista) { print(temp) print(", ") } for(temp in lista)
Run: P14.kts
P14: Duplicate the elements of a list.
Please insert a list of elements, for example: abcd
DesarrolloAplicacionesMoviles
The initial list is: D, e, s, a, r, r, o, l, l, o, A, p, l, i, c, a, c, i, o, n, e, s, M, o, v, i, l, e, s,
The duplicated list is: D, D, e, e, s, s, a, a, r, r, r, r, o, o, l, l, l, l, o, o, A, A, p, p, l, l, i, i, c, c, a, a, c, c, i, i, o, o, n, n, e, e, s, s, M, M, o, o, v, v, i, i, l, l, e, e, s, s,
Process finished with exit code 0
TODO Terminal Build 6: Logcat 4: Run
Event Log
Gradle build finished in 1 s 20 ms (moments ago)
8:1 CRLF UTF-8 4 spaces
Escribe aquí para buscar
ESP
11:29 p. m.
24/02/2020

Respuesta del problema 39

El **pseudocódigo** utilizado es:

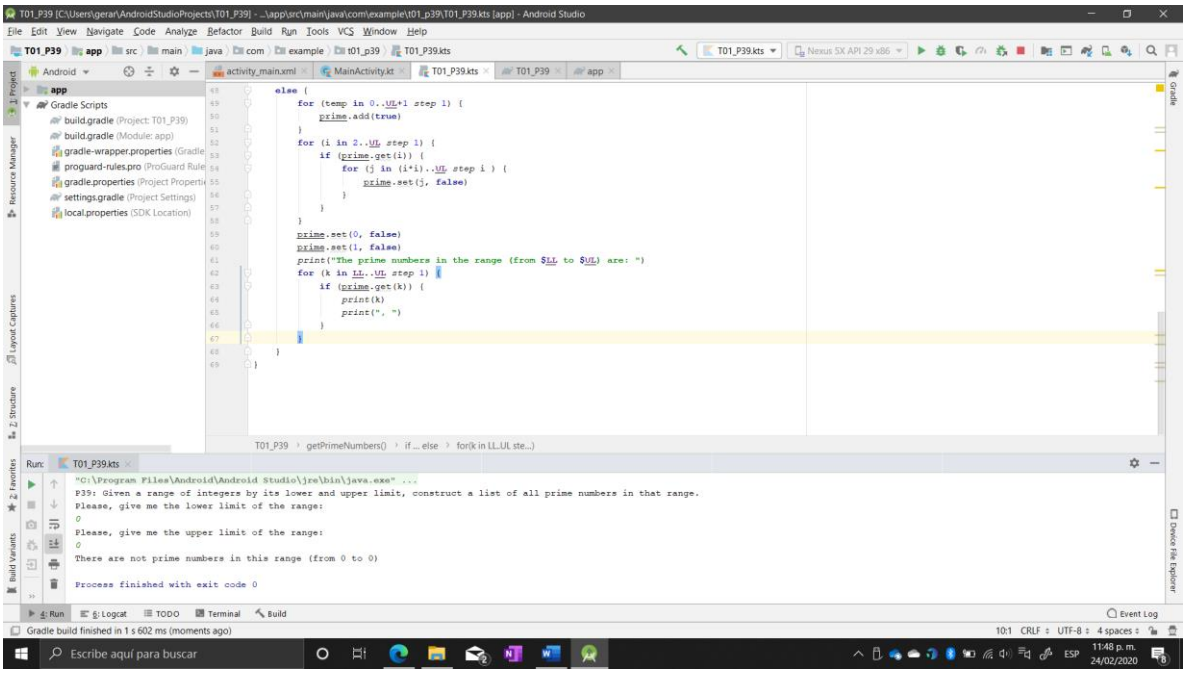
1) Solicitar el rango inferior en consola.
    a. Verificar que el dato ingresado sea numérico.
2) Solicitar el rango superior en consola.
    a. Verificar que el dato ingresado sea numérico.
3) Verificar que el rango superior sea un número mayor al rango inferior.
4) Imprimir el rango dado por el usuario.
5) Resolver utilizando el algoritmo "cibra de Eratóstenes" para detectar los números primos:
    a. Verificar que el rango superior sea menor a dos y si sí:
        i. Imprimir en pantalla "no existen números primos en este rango" y terminar.
    b. Verificar que el rango superior sea 2 y si sí:
        i. Imprimir en pantalla que es el único número primo dentro del rango.
    c. En cualquier otro caso realizar:
        i. Hacer un ciclo *for* desde $i = 2$ (primer número primo) hasta la raíz cuadrada del rango superior:
            1. Crear un *arraylist* de tipo booleano desde 0 hasta el rango superior dado, con todos sus valores en $true$.
            2. Verificar que el número actual no haya sido cambiado de estado.
            3. Iniciar un segundo ciclo *for* desde $j = i$ hasta $\frac{rango\ superior}{i}$:
                a. Cambiar de estado el número $i * j$.
6) Imprimir los números primos dentro del rango dado.

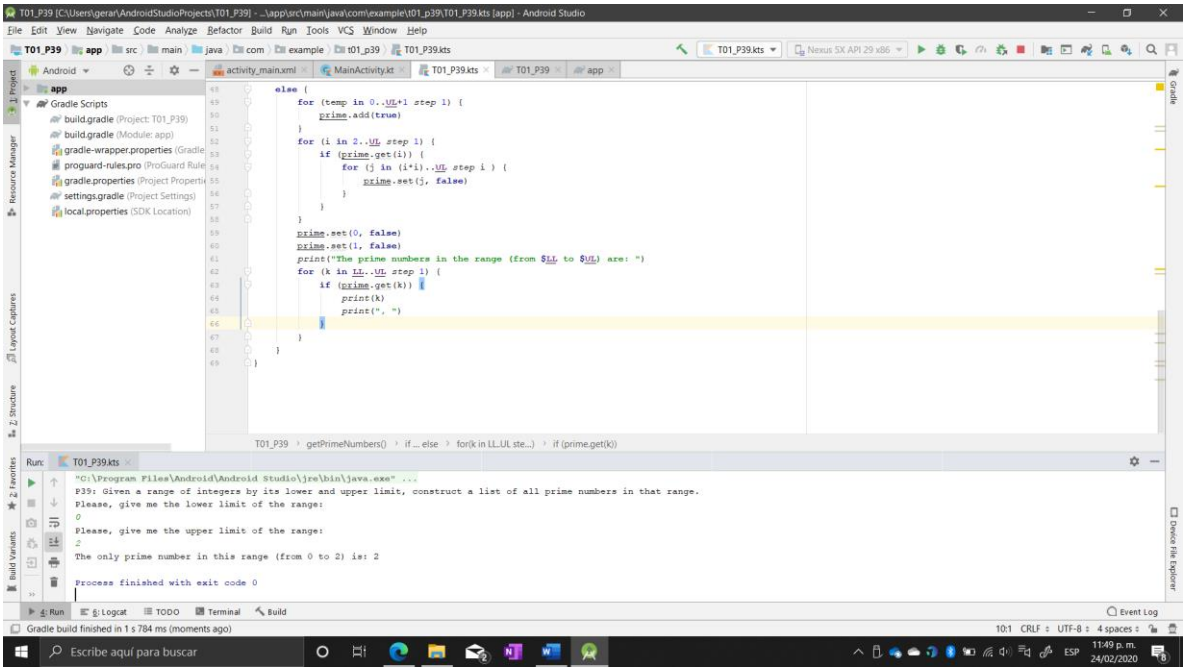
El **código** utilizado es el siguiente:

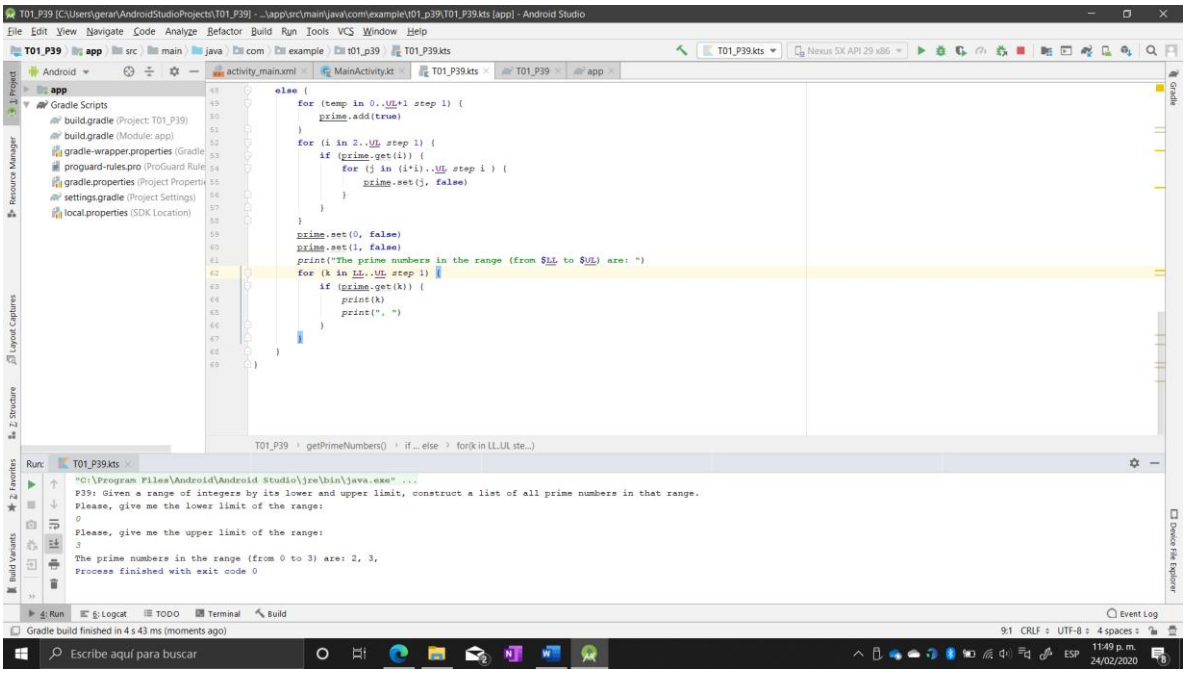
```
package com.example.t01_p39
// Gerardo Daniel Naranjo Gallegos, A01209499
println("P39: Given a range of integers by its lower and upper
limit, construct a list of all prime numbers in that range.")
var repeat = true
var LL = 0
var UL = 0
do {
    println("Please, give me the lower limit of the range: ")
    var lowerLimit = readLine()!!
    try {
        LL = lowerLimit.toInt()
    } catch (e: NumberFormatException) {
        println("Please, give a valid number, like: 0")
        break
    }
    println("Please, give me the upper limit of the range: ")
    var upperLimit = readLine()!!
    try {
        UL = upperLimit.toInt()
    } catch (e: NumberFormatException) {
```

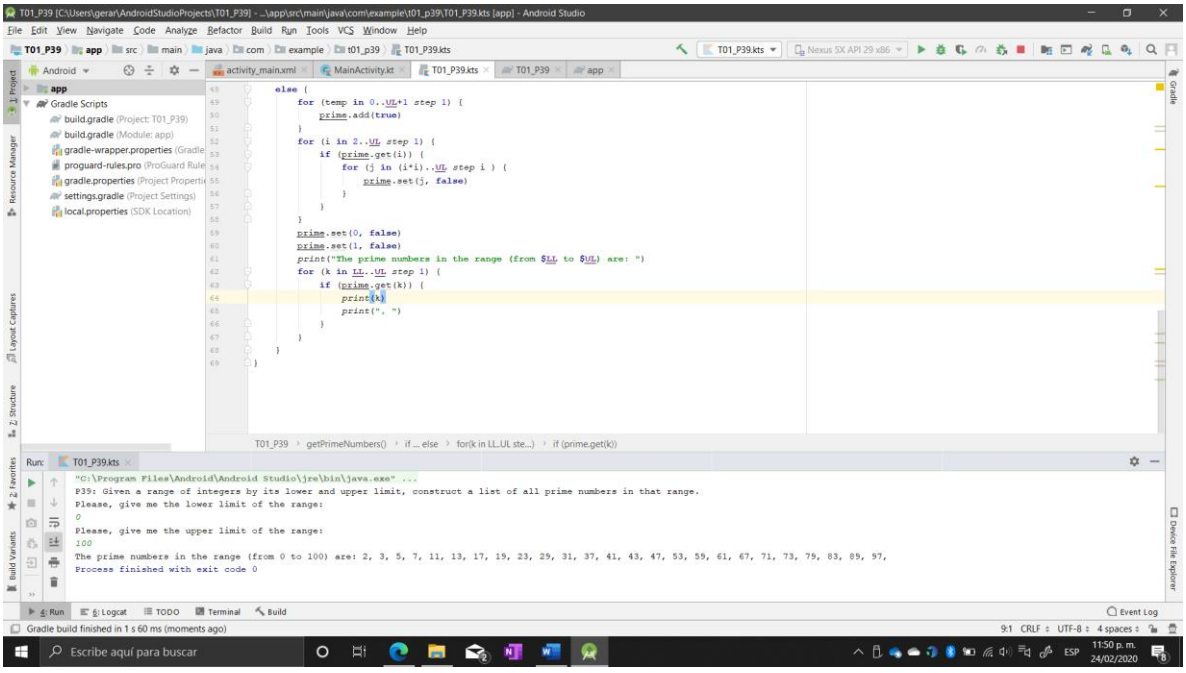
```
        println("Please, give a valid number, like: 1")
        break
    }
    if (LL <= UL) {
        repeat = false
        getPrimeNumbers()
    }
    else {
        println("Please, give a valid range.")
    }
} while (repeat)
// Solved using Sieve of Eratosthenes:
https://www.geeksforgeeks.org/sieve-of-eratosthenes/
fun getPrimeNumbers () {
    var prime:ArrayList<Boolean> = ArrayList()
    if (UL < 2) {
        println("There are not prime numbers in this range (from
$LL to $UL)")
        return
    }
    else if (UL == 2) {
        println("The only prime number in this range (from $LL to
$UL) is: 2")
        return
    }
    else {
        for (temp in 0..UL+1 step 1) {
            prime.add(true)
        }
        for (i in 2..UL step 1) {
            if (prime.get(i)) {
                for (j in (i*i)..UL step i ) {
                    prime.set(j, false)
                }
            }
        }
        prime.set(0, false)
        prime.set(1, false)
        print("The prime numbers in the range (from $LL to $UL)
are: ")
        for (k in LL..UL step 1) {
            if (prime.get(k)) {
                print(k)
                print(", ")
            }
        }
    }
}
```

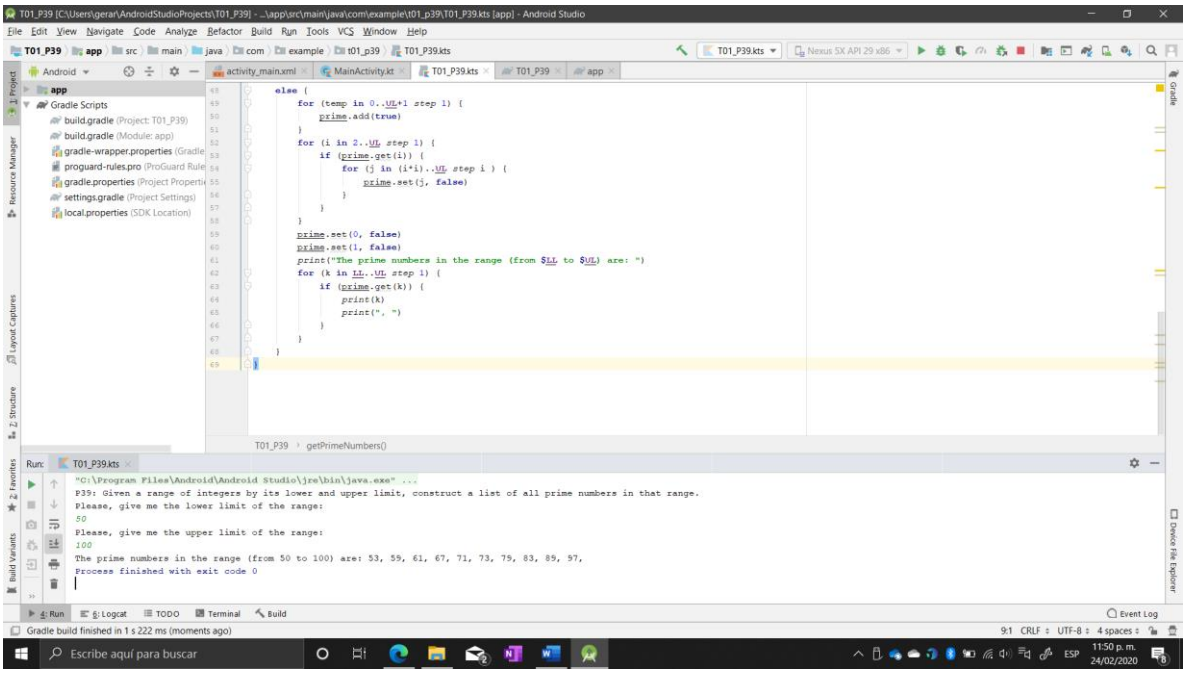
Respecto a los **casos de prueba**, se utiliza el dado por el problema (rango de 7 a 31), así como los siguientes rangos aportados por el alumno: 0 a 0, 0 a 2, 0 a 3, 0 a 100 y 50 a 100. Además, dos últimos casos de prueba para comprobar la validación de recibir números y de recibir un rango correcto.

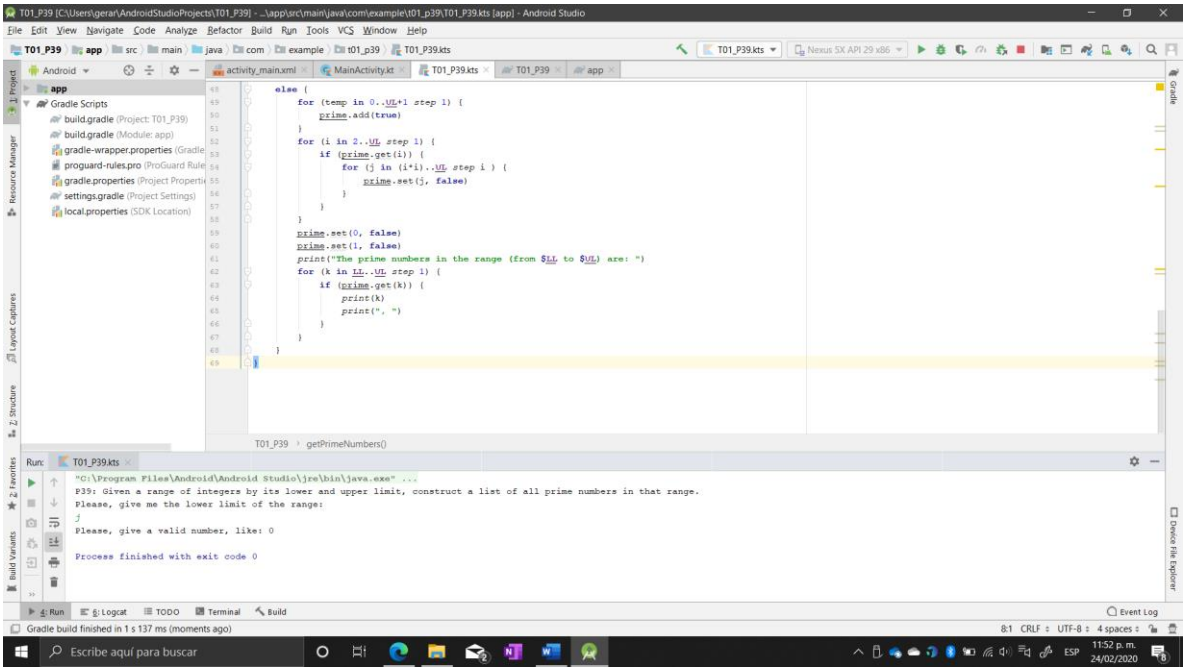


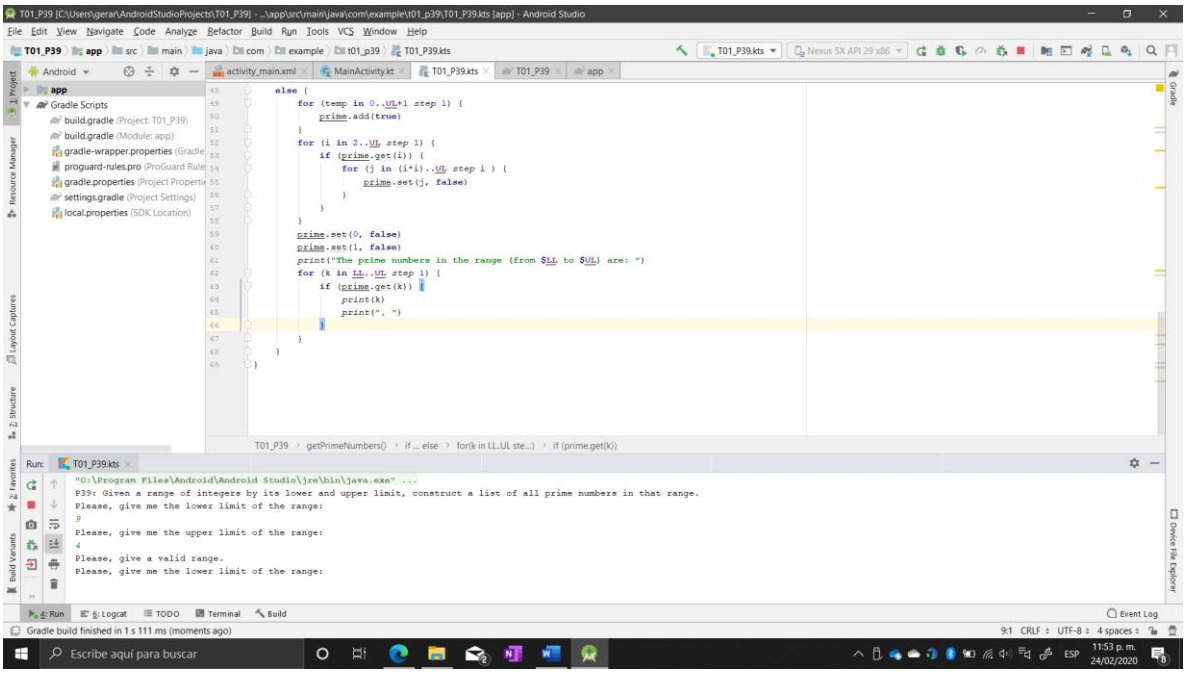



T01_P39 [C:\Users\gerar\AndroidStudioProjects\T01_P39] - ...\app\src\main\java\com\example\t01_p39\T01_P39.kts [app] - Android Studio
File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
T01_P39 app src main java com example t01_p39 T01_P39.kts
activity_main.xml MainActivity.kt T01_P39.kts T01_P39 app
else {
for (temp in 0..UL+1 step 1) {
prime.add(true)
}
for (i in 2..UL step 1) {
if (prime.get(i)) {
for (j in (i*i)..UL step i) {
prime.set(j, false)
}
}
}
prime.set(0, false)
prime.set(1, false)
print("The prime numbers in the range (from $LL to $UL) are: ")
for (k in LL..UL step 1) {
if (prime.get(k)) {
print(k)
print(", ")
}
}
}
}
T01_P39 getPrimeNumbers() if ... else for(k in LL..UL ste...)
Run: T01_P39.kts
"C:\Program Files\Android\Android Studio\jre\bin\java.exe" ...
P39: Given a range of integers by its lower and upper limit, construct a list of all prime numbers in that range.
Please, give me the lower limit of the range:
0
Please, give me the upper limit of the range:
3
The prime numbers in the range (from 0 to 3) are: 2, 3,
Process finished with exit code 0
4: Run 6: Logcat TODO Terminal Build
Event Log
Gradle build finished in 4 s 43 ms (moments ago)
9:1 CRLF UTF-8 4 spaces
Escribe aquí para buscar
ESP 11:49 p. m. 24/02/2020


T01_P39 [C:\Users\gerar\AndroidStudioProjects\T01_P39] - ...\app\src\main\java\com\example\t01_p39\T01_P39.kts [app] - Android Studio
File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
T01_P39 app src main java com example t01_p39 T01_P39.kts
activity_main.xml MainActivity.kt T01_P39.kts T01_P39 app
else {
for (temp in 0..UL+1 step 1) {
prime.add(true)
}
for (i in 2..UL step 1) {
if (prime.get(i)) {
for (j in (i*i)..UL step i) {
prime.set(j, false)
}
}
}
prime.set(0, false)
prime.set(1, false)
print("The prime numbers in the range (from $LL to $UL) are: ")
for (k in LL..UL step 1) {
if (prime.get(k)) {
print(k)
print(", ")
}
}
}
}
T01_P39 getPrimeNumbers() if ... else for(k in LL..UL ste...) if (prime.get(k))
Run: T01_P39.kts
"C:\Program Files\Android\Android Studio\jre\bin\java.exe" ...
P39: Given a range of integers by its lower and upper limit, construct a list of all prime numbers in that range.
Please, give me the lower limit of the range:
0
Please, give me the upper limit of the range:
100
The prime numbers in the range (from 0 to 100) are: 2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41, 43, 47, 53, 59, 61, 67, 71, 73, 79, 83, 89, 97,
Process finished with exit code 0
4: Run 6: Logcat TODO Terminal Build
Event Log
Gradle build finished in 1 s 60 ms (moments ago)
9:1 CRLF UTF-8 4 spaces
Escribe aquí para buscar
ESP 11:50 p. m. 24/02/2020


T01_P39 [C:\Users\gerar\AndroidStudioProjects\T01_P39] - ...\app\src\main\java\com\example\t01_p39\T01_P39.kts [app] - Android Studio
File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
T01_P39 app src main java com example t01_p39 T01_P39.kts
activity_main.xml MainActivity.kt T01_P39.kts T01_P39 app
Android
app
Gradle Scripts
build.gradle (Project: T01_P39)
build.gradle (Module: app)
gradle-wrapper.properties (Gradle
proguard-rules.pro (ProGuard Rule
gradle.properties (Project Properti
settings.gradle (Project Settings)
local.properties (SDK Location)
else {
for (temp in 0..UL+1 step 1) {
prime.add(true)
}
for (i in 2..UL step 1) {
if (prime.get(i)) {
for (j in (i*i)..UL step i) {
prime.set(j, false)
}
}
}
prime.set(0, false)
prime.set(1, false)
print("The prime numbers in the range (from $LL to $UL) are: ")
for (k in LL..UL step 1) {
if (prime.get(k)) {
print(k)
print(", ")
}
}
}
}
T01_P39 getPrimeNumbers()
Run: T01_P39.kts
"C:\Program Files\Android\Android Studio\jre\bin\java.exe" ...
P39: Given a range of integers by its lower and upper limit, construct a list of all prime numbers in that range.
Please, give me the lower limit of the range:
50
Please, give me the upper limit of the range:
100
The prime numbers in the range (from 50 to 100) are: 53, 59, 61, 67, 71, 73, 79, 83, 89, 97,
Process finished with exit code 0
4: Run 6: Logcat TODO Terminal Build
Event Log
Gradle build finished in 1 s 222 ms (moments ago)
9:1 CRLF UTF-8 4 spaces
Escribe aquí para buscar
ESP
11:50 p. m.
24/02/2020


T01_P39 [C:\Users\gerar\AndroidStudioProjects\T01_P39] - ...\app\src\main\java\com\example\t01_p39\T01_P39.kts [app] - Android Studio
File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
T01_P39 app src main java com example t01_p39 T01_P39.kts
activity_main.xml MainActivity.kt T01_P39.kts T01_P39 app
Android
app
Gradle Scripts
build.gradle (Project: T01_P39)
build.gradle (Module: app)
gradle-wrapper.properties (Gradle
proguard-rules.pro (ProGuard Rule
gradle.properties (Project Properti
settings.gradle (Project Settings)
local.properties (SDK Location)
else {
for (temp in 0..UL+1 step 1) {
prime.add(true)
}
for (i in 2..UL step 1) {
if (prime.get(i)) {
for (j in (i*i)..UL step i) {
prime.set(j, false)
}
}
}
prime.set(0, false)
prime.set(1, false)
print("The prime numbers in the range (from $LL to $UL) are: ")
for (k in LL..UL step 1) {
if (prime.get(k)) {
print(k)
print(", ")
}
}
}
}
T01_P39 getPrimeNumbers()
Run: T01_P39.kts
"C:\Program Files\Android\Android Studio\jre\bin\java.exe" ...
P39: Given a range of integers by its lower and upper limit, construct a list of all prime numbers in that range.
Please, give me the lower limit of the range:
j
Please, give a valid number, like: 0
Process finished with exit code 0
4: Run 6: Logcat TODO Terminal Build
Event Log
Gradle build finished in 1 s 137 ms (moments ago)
8:1 CRLF UTF-8 4 spaces
Escribe aquí para buscar
ESP
11:52 p. m.
24/02/2020



Referencias

dkandalov. (2020). *Ninety-Nine Kotlin Problems*. Retrieved from GitHub: https://github.com/dkandalov/kotlin-99

Kotlin. (2020). *Basic Syntax*. Retrieved from Kotlin Lang: https://kotlinlang.org/docs/reference/basic-syntax.html

Kotlin. (2020). *Control Flow: if, when, for, while*. Retrieved from Kotlin Lang: https://kotlinlang.org/docs/reference/control-flow.html

Kotlin. (2020). *List*. Retrieved from Kotlin Lang: https://kotlinlang.org/api/latest/jvm/stdlib/kotlin.collections/-list/index.html

Kotlin. (2020). *Ranges and Progressions*. Retrieved from Kotlin Lang: https://kotlinlang.org/docs/reference/ranges.html

Mithun Kumar, Radhesh Sarma & Shivam Kumar Nayak. (2020). *Sieve of Eratosthenes*. Retrieved from Geeks for Geeks: https://www.geeksforgeeks.org/sieve-of-eratosthenes/

Parewa Labs Pvt. Ltd. (2020). *Kotlin Basic Input/Output*. Retrieved from Programiz: https://www.programiz.com/kotlin-programming/input-output

Parewa Labs Pvt. Ltd. (2020). *Kotlin Program to Check if a String is Numeric*. Retrieved from Programiz: https://www.programiz.com/kotlin-programming/examples/check-string-numeric