**Day 1**

1-if you don’t want to count wrong answer, you can use (try and catch) inside while loop and add the incrementation at the end of (try)

So, if there is any mistake the try will crash before the incrementation happen.

Ex:

while(a<3) {

try {

User should Enter Number

If user enter String, the program will go to catch, and won’t do the next line

a++

}

catch {

}

}

2-you can use (when) in this way:

when {

A==5 -> do something

10<b<0 -> out of range

else -> do something

}

3-I tried with Var Str=readLine():

1. Str.equals(“STRING”)
2. Str == “STRING”

And it all works

4-It is hard to make (for loop) infinite. So instead, if you want infinite loop you should use (While Loop)

1. for (I in 1..3)
2. While (true)

**Day 2**

1-we could use readLine with String:

* var str=readLine()!!.toString()
* var str=readLine()!!
* var str=readLine().toString()

2-we could check the readLine Entry by using if condition

* if(readLine().isEmpty())
* if(readLine()!=null)

3-The difference between ArrayList and arrayListOf

When we initialize empty 2D Array we can write:

-ArrayList<ArrayList<String>>()

-arrayListOf<ArrayList<String>>()

but we can't put the arrayListOf inside:

ex: arrayListOF<arrayListOF<String>>()

because arrayListOf is function, and inside these brackets <> only accept Type(String) or class(ArrayList)

4-we can use these functions with the list and arrays

* list.sort() 🡺 to sort numbers
* list.size 🡺 to know the length
* list.sum() 🡺 to sum the numbers inside the array

5-we add to the 2D array using

answers.add(*arrayListOf*(“country”,”capital”))

6-i can use print method with 2D array using for loop inside for loop:

for(i in answers) {  
 for (c in i)  
 *print*("$c ")  
 *println*()  
}

or

for(i in answers)  
 *println*("Country ${i[0]} its Capital ${i[1]}")

Don’t Forget To Use The {} inside the String to Print The Array

7-Always use float With numbers 😊

8-to enter char (single char – single number) can’t enter two numbers (55)

val c=*readLine*()!!.*single*()

9-Functions Can Be Written outsit the main. Anywhere before or after

Functions also can be Written inside the main, However you should write the function on the top of the main Before you call it. Because if you write the function last, and you call it first there will be an error because the compiler couldn’t compile the function yet.