1. **What is Inheritance? Try to explain in Mandarin.**

Inheritance中文是繼承，它在使用上主要就是用來繼承class的一些屬性以及能力等等．  
跟他相關的幾個重點：  
 (1) 所有class預設都是final，也就是無法繼承．  
 (2) 如果我希望class可以繼承，需要在class前面打上open，才能讓其他的class繼承，

而被打上open class的我們會稱呼為“父”class．

(3) 繼承者為subclass通常會使用：來繼承，並且在其中通常會用override來覆寫“父”

class的一些功能或者能力

為何會使用繼承，因為往往在設定上會有一些重複性功能是大家都通用的，通常會以它來作為“父”class，而其餘的子類別（subclass），則可以再依狀況去進行改變．  
以下舉例：  
在遊戲中，基礎能力值是每個職業都會有的，像是HP, MP, 攻擊, 防禦，這些基礎能力由於每個職業都會有，所以他會是父“class”，但是依照各個職業的不同，可能法師還會有SP之類的，所以法師這個subclass除了有“父”class的基礎能力之外，他還會依照法師獨有的特性有其他的能力或者功能，所以繼承父class並依照所需在寫成自己的subclass法師．

**2. How to split a string in Android? Explain your understanding, and even better to attach some sample code.**

使用split(“並在這邊可以設立條件”)這個函數，舉例來說

(1)最常見的例子:

val str = "String"

val divide = str.split("")

println(divide) //輸出為[, S, t, r, i, n, g, ]

如果我在split()後面的括號沒有進行任何設定，則他就會將上述string，一個一個做切割，並以陣列的方式回傳給我們．

(2)上述衍生的例子:

val str = "String"

val divide = str.split("i")

println(divide) //輸出為[Str, ng]

代表的意思就是說我希望以“i“作為切割基礎，所以他就會從i那邊切成兩部分以陣列的方式回傳給我．

通常split()後面條件的設定依照狀況不同，可以使用各種用法達到分割．

3. What kind of key-value pair collection do we use in Android? Attach a sample code that uses for-loop to print each key-value pair.

在kotlin中我們會使用map來進行key-value的操作，map通常有好幾種用法:

1. 直接建立一個不可更改的靜態map: mapOf:

val map = mapOf(1 to "A", 2 to 3)

val map =mapOf(Pair(1, “A”), Pair(2, 3))

2. 建立一個動態的map, 使用mutableMap: mutableMapOf:

var mutablemap = mutableMapOf<Int, String>(1 to "A", 2 to "B")

mutablemap.put(3 ,"C")

mutablemap[4] = "Ａ"

for ((k, v) in mutablemap) {

println("Key: $k, Value: $v")

}

Result:

Key: 1, Value: A

Key: 2, Value: B

Key: 3, Value: C

Key: 4, Value: Ａ

上述各用了兩種方式來建立或者加入key-value值，有幾個重點比較需要注意，第一個是key值是單一不能重複，但是value值是可以重複的；第二是，mutablemap中，我們可以自己不用宣告型態（像是：string或者Int)，只要後面有例子，IDE會自動幫我們判定我們所使用的型態為何，但是假如我們在mutableMapOf，後面沒有舉例的話，就一定需要主動宣告型態，並且宣告後後續就不能改動．

PS: 由於mapOf是一個靜態不可更改的map，所以他的key-value的型態，組之間可以不一樣，因為他只能讀取不能更改．

4. What’s the difference between ArrayList and LinkedList? Try to explain in Mandarin.

就幾個面向去做討論

(1)數據結構：

ArrayList是透過數組去做決定的，數組代表的意義就是，他是利用“連續”的空間來儲存元素，所以非常看重每一個值得index(索引），藉由index叫出資料值是非常快速的．

LinkedList是透過雙向鏈表去實現的，雙向鏈表代表它是由多個節點(Node)組成，每一個節點都包含了三部分1. Item內容 2. Prev前引用 3. Next後引用．所以如果要叫出特定某個值，每次都會需要遍歷整個list來找出值．

(2)查找：  
 ArrayList由於是透過“索引”（順序）來進行資料的建構，因此要找特定資料時，透過index可以直接找出內存中的位置，存取的效能會較好且速度較快．

LinkedList由於是透過每一個值得節點所建構的，因此找特定資料時，需要透過遍歷整個list每個節點，進而找出特定節點的值，存取的效能跟速度會比較差．

(3)插入刪除：

ArrayList由於是透過數組建立的，它對於一開始宣稱的長度（容量）是“固定”的，如果說要新增的話，有可能因為容量不夠導致需要擴容，並且由於它是以index為基準，所以在新增或刪除中間值的時候，對於整體程式在跑的效能會差很多．  
 LinkedList是透過雙向鏈表的節點建立的，由於在儲存上他看重的是每一個節點的前後，並非順序，因此我想要在中間新增或刪除時，只要找到對應節點去操作，速度以及效能都是非常優於ArrayList．

5. What are Value Type variables? List out the Value Type variables you know.

講到Value Type就必須先跟Reference Type一起做比較，Value Type變數會直接儲存變數資料的值，例如 a = 10且b = a，在記憶體中他會將變數b的值直接儲存為10；而reference Type則是儲存變數資料的“位址”，例如a = “Tom”儲存在記憶體的位置是0xabcd , b = a再產生值得過程中，他是依照位址0xabcd去取值，而並非直接複製他的值．  
常見的Value Type variables：Byte, Short, Int, Long, Float, Double, Boolean, Char．

6. What’s the maximum value of Int? If you want to represent integers larger than the limit, what should you do instead of using int variables?

Int這型別中，最大值是2147483647．如果要使用超過這數值的話，需要將型別改成Long，Long的最大值為9223372036854775807．

整數部分總共有四種，每個上限值都不一樣，主要是因為吃的記憶體數不同，所以上限值也不同．

型別 記憶體 最大值 最小值  
Byte 8bit 127 -128

Short 16bit 32767 -32768

Int 32bit 2147483647 -2147483648

Long 64bit 9223372036854775807 -9223372036854775808