

QUIZ Sec02 (Time 1 hr.) คนที่ทำเสร็จแล้วขอให้ยังออนไลน์ใน zoom อยู่ด้วยดีใจจนหมดเวลาละ

- ฟังก์ชันต่าง ๆ ให้เขียนแบบ **Recursive** เท่านั้น ห้ามใช้ loop ถ้าไม่เขียนด้วย **recursion** จะได้ 0 คะแนนในข้อนั้น ๆ
- อนุญาตให้ใช้ เมธอดของลิสต์ได้แค่ **isEmpty, length, head, tail, ::, ++** เท่านั้น ใครใช้เกินมา จะได้ 0 คะแนนในข้อนั้น ๆ
- เขียนเมธอดใหม่เองจากเมธอดพื้นฐานที่อนุญาตข้างต้นได้
- ให้แยกหนึ่งข้อต่อหนึ่งไฟล์ ตั้งชื่อไฟล์ตามข้อ เช่น **Question01.scala**
- ในแต่ละข้อให้เขียน **main** เพื่อทดสอบได้ตามใจ อาจารย์จะตรวจโดยใช้ **main** ของอาจารย์เอง
- การส่ง ส่ง โดย **zip** ไฟล์ของทั้งสองข้อรวมกัน แล้วตั้งชื่อ zip file เป็น **ID_scalaQuiz** เช่น **6332011421_scalaQuiz.zip**

1. (4 คะแนน) จงเขียนฟังก์ชัน `def applySeq(x:Int) (fList: List[Int=>Int]): Int = {` ซึ่งได้คำตอบเป็นค่า `int` ที่เกิดจากการเอา ฟังก์ชันในลิสต์ `fList` ไป `apply` กับ `x` แล้วเอาผลเป็นอินพุตของฟังก์ชันต่อไปในลิสต์เรื่อย ๆ

ตัวอย่างเช่น

```
println(applySeq(1)(List((x => x+1), (x => x*x), (x => x+5))))
```

จะได้คำตอบจาก เอา 1 ไปเป็นอินพุตของ `x => x+1` ดังนั้นจึงได้ 2

เอา 2 ไปเป็นอินพุตของ `x => x*x` ได้ 4

เอา 4 ไปเป็นอินพุตของ `x => x+5` ได้ 9 ซึ่งเป็นคำตอบสุดท้าย

```
println(applySeq(100)(List()))
```

 จะได้ 100 เป็นคำตอบ

2. (8 คะแนน) จงเขียนฟังก์ชัน `moreThanHalf(l1:List[Any],l2:List[Any]):Boolean = {` ให้ถือว่า

- l1 กับ l2 มีขนาดเท่ากันเสมอ
- l1 ไม่มีข้อมูลซ้ำกันภายใน
- l2 ไม่มีข้อมูลซ้ำกันภายใน

ฟังก์ชันนี้ ทดสอบว่า มีของมากกว่าครึ่งหนึ่งของจำนวนของใน l1 หรือเปล่า ที่อยู่ใน l2 ด้วย (รีเทิร์น true หรือ false)

ตัวอย่างเทสเคสและคำตอบที่ควรได้ เป็นดังนี้

```
val l00 = List()
val l01 = List()
val l1 = List(1,4,3,2,7,8)
val l2 = List(2,9,7,1,6,10)
val l3 = List(12,11,6,5,3,2)
val l4 = List("1","2","3","4","5")
val l5 = List("9","7","5","3","1")
val l6 = List("2","4","6","8","10")
println(moreThanHalf(l00,l01)) //false
println(moreThanHalf(l1,l01)) //false
println(moreThanHalf(l1,l2)) //true เพราะจาก l1 มี 1,2,7 อยู่ใน l2 ด้วย (ครึ่งหนึ่ง สามตัวพอดี)
println(moreThanHalf(l1,l3)) //false จาก l1 มีแค่ 3 กับ 2 ที่อยู่ใน l3 ด้วย (แค่สองตัว)
println(moreThanHalf(l4,l5)) //true จาก l4 มี "1","3","5" อยู่ใน l5 ซึ่งเกินครึ่ง (2.5 ตัว)
println(moreThanHalf(l4,l6)) //false จาก l4 มี "2","4" เท่านั้นที่อยู่ใน l6 จึงไม่ถึงครึ่งนี้
```