Quiz (60 mins)

- สร้างโฟลเดอร์ c:\temp\proglangQuiz01_(seat no.)_(ID)_(Firstname) ขึ้นมา ถ้ายังไม่ได้ทำ
    - ตัวอย่าง สำหรับคนนั่งโต๊ะหมายเลข 1: C:\temp\ proglangQuiz01_01_6510000021_Amorn
- ในโฟลเดอร์ c:\temp\proglangQuiz01_(seat no.)_(ID)_(Firstname) ใช้ IntelliJ new project ชื่อ Q1 กับ Q2 ขึ้นมา (ทำทีละโปรเจ็ค)
- จากนั้นในโปรเจ็ค ให้เอาไฟล์โจทย์ .scala ที่โหลดใน MyCourseville copy เข้าโฟลเดอร์ src\main\scala ของแต่ละโปรเจ็ค
- ฟังก์ชั่นต่าง ๆ ให้เขียนแบบ Recursive เท่านั้น ห้ามใช้ลูป ถ้าไม่เขียนด้วย recursion จะได้ 0 คะแนนในข้อนั้น ๆ
- อนุญาต ให้ใช้ เมธอดของลิสต์ ได้แค่ isEmpty, length, head, tail, ::, ++ เท่านั้น ใครใช้เกินมา จะได้ 0 คะแนนในข้อนั้น ๆ
- อนุญาตให้สร้างลิสต์ โดยใช้ List(สมาชิก1,สมาชิก2,...) ได้
- ไม่อนุญาตให้ access ลิสต์ด้วย index
- เขียนเมธอดใหม่เองจากเมธอดพื้นฐานที่อนุญาตข้างต้นได้
- ควิสนี้มีสองข้อ ให้แยกหนึ่งข้อต่อหนึ่งโปรเจ็ค ตั้งชื่อไฟล์หลักตามข้อ Q1.scala, Q2.scala ชื่อคลาสก็ต้องเหมือนชื่อไฟล์ ถ้าไม่ทำตามนี้จะได้ 0 คะแนน (ทำมาให้แล้ว อย่าไปเปลี่ยนชื่อมันก็แล้วกัน)
- อาจารย์จะตรวจโดยใช้ main ของอาจารย์เอง ที่อยู่ในอีกไฟล์หนึ่ง (มีไฟล์ main ตัวอย่างให้ดู ในไฟล์ TestQ1, TestQ2)
- การส่ง เปิด IntelliJ ทิ้งไว้ แล้วทางศูนย์คอมจะดำเนินการเซฟไฟล์เอง

1. (5 คะแนน) จงเขียนฟังก์ชั่น

```
def sortTwoLists(l1:List[Int],l2:List[Int]):(List[Int],List[Int]) = {
```

ฟังก์ชั่นนี้รับลิสต์สองลิสต์ แล้วรีเทิร์น tuple (a,b) ที่มีสองลิสต์ (ขนาดเท่า l1 และ l2 ตามลำดับ โดย ลิสต์ l1 กับ l2 สามารถเป็นลิสต์ว่างได้) โดยลิสต์ทั้งสองลิสต์ที่รีเทิร์นออกมาจะต้องมีสมาชิกเรียงจากน้อยไปมาก และสมาชิกใน a ต้องเรียงมาก่อนสมาชิกใน b

ยกตัวอย่างเช่น

ถ้า l1 เป็น List(5,2,4) และ l2 เป็น List(7,8,9,1,2,3,5)

การเรียก sortTwoLists(l1,l2) จะได้ (List(1,2,2),List(3,4,5,5,7,8,9)) เป็นคำตอบ จะเห็นได้ว่า

- ลิสต์แรกมีจำนวนสมาชิก 3 ตัว เท่ากับ l1
- ลิสต์ที่สองมีสมาชิก 7 ตัว เท่ากับ l2
- ทั้งสองลิสต์ สมาชิกจะเรียงกัน จากน้อยไปมาก โดยค่าน้อยจะเรียงในลิสต์แรกก่อน

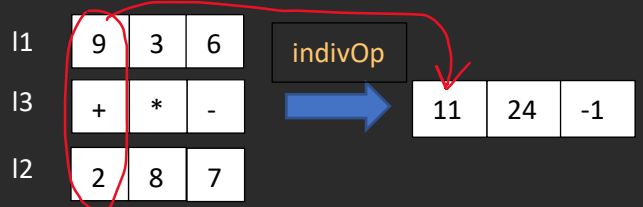
ดูกรณีตัวอย่างอื่น ได้ในไฟล์ TestQ1.scala

2. (5 คะแนน) จงเขียนฟังก์ชั่นต่อไปนี้

```
def indivOp(l1:List[Int], l2: List[Int], l3:List[(Int,Int) => Int]):List[Int]
= {
```

ฟังก์ชั่นนี้มีโค้ดมาให้แล้ว แต่ว่า ไม่ได้เป็น tail recursion ให้เปลี่ยนโค้ดภายใน ให้ใช้ tail recursion ถ้าไม่เปลี่ยนจะไม่ได้คะแนน

ฟังก์ชั่นนี้ รับลิสต์ของจำนวนเต็มสองลิสต์และลิสต์ของฟังก์ชั่นอีกหนึ่งลิสต์ (ทั้งสามลิสต์มีจำนวนสมาชิกเท่ากัน และแต่ละลิสต์สามารถเป็นลิสต์ว่างได้) ทำการเรียกใช้ฟังก์ชั่นดังรูปตัวอย่าง (สมมุติให้ลิสต์มีสมาชิก 3 ตัว) แล้วรีเทิร์นผลลัพธ์เป็นลิสต์ของจำนวนเต็ม



ตัวอย่างการทำงาน ดูได้ในไฟล์ TestQ2.scala