

## QUIZ (Time 60 minutes.)

- สร้างไฟล์เดอร์ `c:\temp\proglangQuiz02_(seat no.)(ID)(Firstname)` ขึ้นมา ถ้ายังไม่ได้ทำ
- ตัวอย่าง สำหรับคนนั่งโต๊ะหมายเลข 1: `C:\temp\proglangQuiz02_01_6510000021_Amorn`
- ในไฟล์เดอร์ `c:\temp\proglangQuiz02_(seat no.)(ID)(Firstname)` ใช้ IntelliJ new project ชื่อ Q1 กับ Q2 ขึ้นมา(ทำทีละโปรเจกต์) ต้องตั้งชื่อตามนี้เป๊ะๆ ไม่งั้นตัวตรวจให้คะแนนจะไม่ตรวจให้ ต้องสร้าง scala project นะ อย่าไปสร้าง Java project
- ในโปรเจกต์ Q1 ให้สร้าง package q1 ส่วนในโปรเจกต์ Q2 ก็สร้าง package q2



- จากนั้น ให้เอาไฟล์โจทย์ .scala ที่โหลดใน MyCourseville copy เข้า package ของแต่ละโปรเจกต์ จะเป็นไฟล์ ชื่อ Q1.scala และ Q2.scala . Q1.scala ให้เอาใส่ โปรเจกต์ Q1 ส่วน Q2.scala ก็ให้เอาใส่โปรเจกต์ Q2
- ฟังก์ชันต่าง ๆ ให้เขียนแบบ Recursive เท่านั้น ห้ามใช้ loop ถ้าไม่เขียนด้วย recursion จะได้ 0 คะแนนในข้อนั้น ๆ
- อนุญาต ให้ใช้ เมธอดของลิสต์ได้แค่ isEmpty, length, head, tail, ::, ++ เท่านั้น ใครใช้เกินมา จะได้ 0 คะแนนในข้อนั้น ๆ
- อนุญาตให้สร้างลิสต์ โดยใช้ List(สมาชิก1,สมาชิก2,...) ได้
- เขียนเมธอดใหม่เองได้
- ไม่อนุญาตให้ access ลิสต์ด้วย index
- Data structure ที่จะใช้ในโปรแกรม อนุญาตให้ใช้แค่ List เท่านั้น ถ้าใช้อย่างอื่นจะได้ 0 คะแนนในข้อที่ใช้

### 1. (10 คะแนน) ทำใน Project Q1

จงเขียนฟังก์ชัน `def identifyMin (x:Int) (l: List[Int=>Int]): Int => Int` รับ `x` ซึ่งเป็นเลขจำนวนเต็ม และรับ `l` ซึ่งเป็นลิสต์ของฟังก์ชัน (ที่รับเลขจำนวนเต็มแล้วรีটারน์จำนวนเต็ม) ซึ่งได้คำตอบเป็นฟังก์ชัน (เลือกจากในลิสต์ `l` นี้) ที่เมื่อเอาไป apply กับ `x` จะให้คำตอบน้อยที่สุด (ให้ถือว่าในลิสต์จะมีอย่างน้อย 1 ฟังก์ชันเสมอ)

(ค่าที่รีটারน์ คือ ฟังก์ชันใดฟังก์ชันหนึ่ง จาก ลิสต์ `l`)

ตัวอย่างเทสรัน อยู่ในไฟล์ Q1.scala ที่ให้แล้ว

### 2. (10 คะแนน) ทำในโปรเจกต์ Q2

จงเขียนฟังก์ชัน `difference (l1:List[Any],l2:List[Any]):List[Any]`

ให้พยายามใช้ tail recursion เพื่อลด stack frame (ถ้าไม่ใช้จะได้คะแนนอย่างมากแค่ครั้งเดียว)

- ฟังก์ชันนี้ รีটারน์ ลิสต์ที่ประกอบด้วย ของจาก `l1` ที่ไม่อยู่ใน `l2` (ให้ถือว่า `l1` ไม่มีของซ้ำภายใน, `l2` ก็ไม่มีของซ้ำภายในเช่นกัน)

เทสเคสอยู่ใน main ของ Q2.scala แล้ว