

## QUIZ Sec01 (Time 45 minutes.)

- 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
- อนุญาตให้สร้างลิสต์โดยใช้ **List(สมาชิก1,สมาชิก2,...)** ได้
- ไม่อนุญาตให้ **access** ลิสต์ด้วย **index**
- เขียนเมธอดใหม่เองจากเมธอดพื้นฐานที่อนุญาตข้างต้นได้
- ให้แยกหนึ่งข้อต่อหนึ่งไฟล์ ตั้งชื่อไฟล์ตามข้อ เช่น **Question01.scala**
- ในแต่ละข้อให้เขียน **main** เพื่อทดสอบได้ตามใจ อาจารย์จะตรวจโดยใช้ **main** ของอาจารย์เอง
- การส่ง ส่ง โดยเอาไฟล์ของทั้งสองข้อ **zip** ส่งทาง **MyCourseville**

1. (10 คะแนน) จงเขียนฟังก์ชัน `def partition(l:List[Int], f:Int => Boolean):List[List[Int]] = {`  
ฟังก์ชันนี้ รับ `l` ซึ่งเป็นลิสต์ของ `integer` และ รับ `f` ซึ่งเป็นฟังก์ชัน(ที่รับ `Int` แล้วรีเทิร์น `Boolean`) เข้ามา  
รีเทิร์น ลิสต์ของลิสต์(มีสมาชิกแค่สองตัว)

- สมาชิกตัวแรกเป็นลิสต์ที่มีสมาชิกทุกตัวเป็นสมาชิกจาก `l` ที่รัน `f` แล้วเป็นจริง
- สมาชิกตัวที่สองเป็นลิสต์ที่มีสมาชิกทุกตัวเป็นสมาชิกจาก `l` ที่รัน `f` แล้วไม่เป็นจริง

ตัวอย่าง ถ้าเรามีฟังก์ชัน `f1` และ `f2` ดังนี้

```
def f1(x:Int):Boolean = {  
  x%2 == 1  
}
```

```
def f2(x:Int):Boolean = {  
  x*x > 10  
}
```

ตัวอย่าง เมธอด `main` จะได้ output ตามที่คอมเมนต์ดังนี้

```
def main(args: Array[String]): Unit = {  
  val l1 = List(1,2,3,4,5)  
  println(partition(l1,f1))           // List(List(1, 3, 5), List(2, 4))  
  println(partition(l1,f2))           // List(List(4, 5), List(1, 2, 3))  
  println(partition(l1,(x => x==0)))   // List(List(), List(1, 2, 3, 4, 5))  
  println(partition(l1,(x => x<6)))    // List(List(1, 2, 3, 4, 5), List())  
}
```

2. (4 คะแนน) จงเขียนฟังก์ชัน `def compliment(f:Int => Int):Int => Int = {`

ฟังก์ชันนี้ รับ ฟังก์ชัน `f` เข้ามา แล้วรีเทิร์นฟังก์ชัน ที่ทำงานเหมือน `f` แต่เปลี่ยนผลลัพธ์จากค่าบวกเป็นลบ ลบเป็นบวก  
ถ้ามีฟังก์ชัน `f1` ดังนี้

```
def f1(x:Int):Int = {  
  x - 1000  
}
```

ตัวอย่างโค้ดใน `main` และผลลัพธ์จะเป็นดังนี้

```
def main(args: Array[String]): Unit = {  
  println(compliment((x => x*x))(-5)) // -25  
  println(compliment((x => x*x))(3))  // -9  
  println(compliment((x => -1*x*x))(3)) // 9  
  val c = compliment(f1)  
  println(c(3))                      // 997  
  println(c(3000))                   // -2000  
}
```