

Lab Quiz 3 Section B4

Q1

Q1 (25 points): Study the following code to answer the question below

```
class Date(var month: Int, var day: Int) {}
```

```
def q1(): Unit = {  
  var dateList = List(  
    new Date(1, 10),  
    new Date(2, 30),  
    new Date(2, 14),  
    new Date(1, 9),  
    new Date(3, 1)  
  )  
  
  val comparator: (Date, Date) => Int =  
    (a: Date, b: Date) => {  
      if (a.month < b.month) { -1 }  
      else if (a.month > b.month) { 1 }  
      else if (a.day < b.day) { -1 }  
      else if (a.day > b.day) { 1 }  
      0  
    }  
  
  dateList = dateList.sortWith(comparator)  
  
  println(dateList)  
}
```

In what order are the elements of the inventory list printed at the end of the q1() method call?

Q2

Q2 (25 points): Study the following code to answer the question below

```
def funRecursion(n: Int): Int = {  
  if (n >= 10) {  
    5  
  } else {  
    -n + funRecursion(n + 2)  
  }  
}
```

What is returned by the method call *funRecursion*(-5)?

Q3

Q3 (25 points): Study the following code to answer the question below

```
def funnerRecursion(n: Int): Int = {  
  if (n <= 10) {  
    -2  
  } else {  
    n + -funnerRecursion(n-4) - funnerRecursion(n-2)  
  }  
}
```

What is returned by the method call *funnerRecursion(16)*?

Q4

Q4 (25 points): Study the following code to answer the question below

```
class Date(var month: Int, var date: Int) {}

def q4(): Int = {
  var myList: LinkedListNode[Date] = new LinkedListNode[Date](
    new Date(arg1, arg2), null
  )
  myList = myList.prepend(new Date(1, 0))
  myList = myList.prepend(new Date(2, 2))
  myList = myList.prepend(new Date(3, 15))
  myList = myList.prepend(new Date(4, 16))
  myList = myList.prepend(new Date(5, 20))

  val doubleDay: Date => Date = (x: Date) => new Date(x.month, x.date*2)
  myList = myList.map(doubleDay)

  val validDay: Date => Int = (x: Date) => {
    if (x.day <= 31 && x.day > 0) {
      1
    } else {
      0
    }
  }

  myList.sumValidDates(validDay)
}

class LinkedListNode[Date](var value: Date, var next: LinkedListNode[Date]) {

  def sumValidDates(f: Date => Int): Int = {
    if (this.next == null) {
      f(this.value)
    } else {
      f(this.value) + this.next.sumValidDates(f)
    }
  }

  def prepend(a: A): LinkedListNode[A] = {
    new LinkedListNode[A](a, this)
  }

  def map[B](f: A => B): LinkedListNode[B] = {
    val newValue = f(this.value)
    if (this.next == null) {
      new LinkedListNode[B](newValue, null)
    }
  }
}
```

```
    } else {  
        new LinkedListNode[B](newValue, this.next.map(f))  
    }  
}  
  
}
```

What does a call to q4() return?