CSC369-01
Winton Gee
Assignment 4

Question 1

```
// This question wants the count of the numbers that are divisible by 3 from a file
def firstQuestion: Unit = {
  val numbers = Source.fromFile(inputPath_1).getLines().map(_.toInt)
  val divCount = numbers.count(_ % 3 == 0);
  println(s"Count of numbers divisible by 3: $divCount")
}
```

Question 2

```
// This question wants to output the highest data of each date
def secondQuestion: Unit = {
  val hashMap = new mutable.HashMap[String, Int]()
  val data = Source.fromFile(inputPath_2).getLines().map(_.split(", "))

  data.foreach { array =>
    val key = array(0)
    val value = array(1).toInt
    // Only update the hashmap if the value does not exist or new temp is higher
    val storedValue = hashMap.getOrElse(key, 0);
    val updatedValue = Math.max(storedValue, value);
    hashMap += (key -> updatedValue)
  }

  // Output the information
  hashMap.foreach { case (key, value) =>
    println(s"Key: $key, Value: $value")
  }
}
```

Question 3

```
// Example line input: John Back, 23, A, CSC369
// Example line output: John Back, 23, (A, CSC369), (B, CSC366)
def thirdQuestion: Unit = {
  val studentData = Source.fromFile(inputPath_3).getLines().map(_.split(", "))
  val studentClasses = mutable.HashMap.empty[(String, String), mutable.TreeSet[ClassGrade]]

  studentData.foreach { data =>
    val key = (data(0), data(1))
```

```
      val classGrade = new ClassGrade()
      classGrade.set(data(2), data(3))

      // Update the existing set or create new set, the ordering is set when the TreeSet is init
      val gradesSet = studentClasses.getOrElseUpdate(key, scala.collection.mutable.TreeSet
        .empty[ClassGrade](Ordering[(String, String)].on(x => (x.grade, x.name))));
      gradesSet += classGrade

      // Update the classes
      studentClasses.update(key, gradesSet)
    }

    // This will sort the studentClasses by the student's name and id
    val sortedStudentClasses = studentClasses.toSeq.sortBy { case (key, _) => (key._1,
key._2.toInt) }

    // Output the information
    sortedStudentClasses.foreach { case ((name, id), grades) =>
      val classGradeInformation = grades.mkString(",");
      println(s"$name, $id $classGradeInformation")
    }

  }

  class ClassGrade {
    var grade = "";
    var name = "";

    def set(grade: String, name: String) = {
      this.grade = grade;
      this.name = name;
    }

    override def toString: String = s"($grade, $name)"
  }
```