**Group Activity 9; CS 3060**

**Names** of students in your group: Sidney Sanders, Amanda Collert, Quinci Drain, Emily Endlish

Points: 10

**Task 1**: (5 points) Say function *foo* gets a list (named *list1*) of strings as input. Then, in function foo do the following. Test your code of *foo* and present test results.

(a) Build a new list *list2* from *list1* after filtering out strings that start with character “c” **and** end with character “d”.

As an example, say list1 = List(“apple”, “cat”, “carad”, “badminton”)

Then, list2 will be List(“apple”, “cat”, “badminton”)

(b) Build a new list *list3* from *list1* by replacing strings that contain “bad” sub-string with “replaced”.

As an example, say list1 = List(“dsgg”, “cwqjdx”, “csvjqjqkd”, “kbads”)

Then, list3 will be List(“dsgg”, “cwqjdx”, “csvjqjqkd”, “replaced”)

A template for both task 1a and task1b is given below:

object GroupWorkNine {

    def main(args: Array[String]): Unit =

    {

        val list1 = List("apple", "cat", "carad", "badminton")

        val list2 = list1.filter(word => !(word.startsWith("c") && word.endsWith("d")))

        println(list2)

        val list3:List[String]= list1.map( word =>

            if (word.contains("bad")) {

                "replaced"

            }

            else {

                word

            }

            )

        println(list3)

    }

}

> scala task1.scala

List(apple, cat, badminton)

List(apple, cat, carad, replaced)

Hint: For task1a, you may see class activity 2, which has a similar example that you can extend. For task1b, you may need to use String.contains method to check for the sub-string.

**Task 2**: (3 points) Consider the following Scala code. Copy the code to a file *task2.scala*. Now run “*scala task2.scala*”. What will be the output? Give brief explanation: basically fill out the blanks shown below.

val list = List(1,9,7,4)

list.foreach(x => println(x\*x)) // output = ? why/how?

1

81

49

16

Why: It traverses the list printing out each item squared.

How: foreach loop

val list2 = list.map(x => x\*x ) // output = ? why/how?

List(1, 81, 49, 16)

Why: Traverses the list replacing the x with its square.

How: Map

val y = list.exists(x => x > 5) // output = ? why/how?

True

Why: An item within the list is greater than 5.

How: Exists

val z = list.forall(x => x > 5) // output = ? why/how?

False

Why: Not every item in the list is greater than 5.

How: ForAll

val s = list.sortWith((x,y) => x > y) // output = ? why/how?

List(9, 7, 4, 1)

Why: Sorting largest to smallest, sortWith traverses until entire list is sorted.

How: sortWith

println(list.foldLeft(0)(\_+\_)) // output = ? why/how?

21

Why: starts with left most item in the list and continues, this adds the list item by item to (0).

How: foldLeft

**Task 3**: (2 points) Write a Scala function *foo that* accepts a list of strings (as parameter) and prints the max-length string and the min-length string.

def foo (list1:List[String]) =

    {

        var maxLength = ""

        var minLength = ""

        var list2 = list1.sortWith((x,y) => x.size > y.size)

        var first = list2.head

        var last = list2.last

        println(first)

        println(last)

    }

> scala task3.scala

badminton

cat

Hint: You may use higher order function *sortWith*, which is available in scala lecture 2 ppt.

Submission: Submit one copy (per group) on Canvas.