**Amrita School of Engineering**

**Department of Computer Science and Engineering**

**19CSE313 – Principles of programming Languages**

**Lab-Evaluation-2**

**Date: 11/04/2022** **Topic: Scala** **Time: 1hrs**

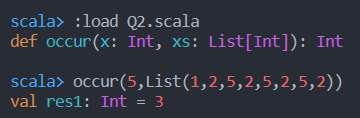
1. **Write a functional Scala program occur that finds the number of occurrences of a given element in a given list. For example occur 5 (1,2,5,2,5,2,5,2) should return 3 as 5 occurs 3 times in the given list. It is expected that your function occur should work on nested lists too. So occur 5 ((1,2,5,2), (5,2),(5,2)) should still return 3!!**

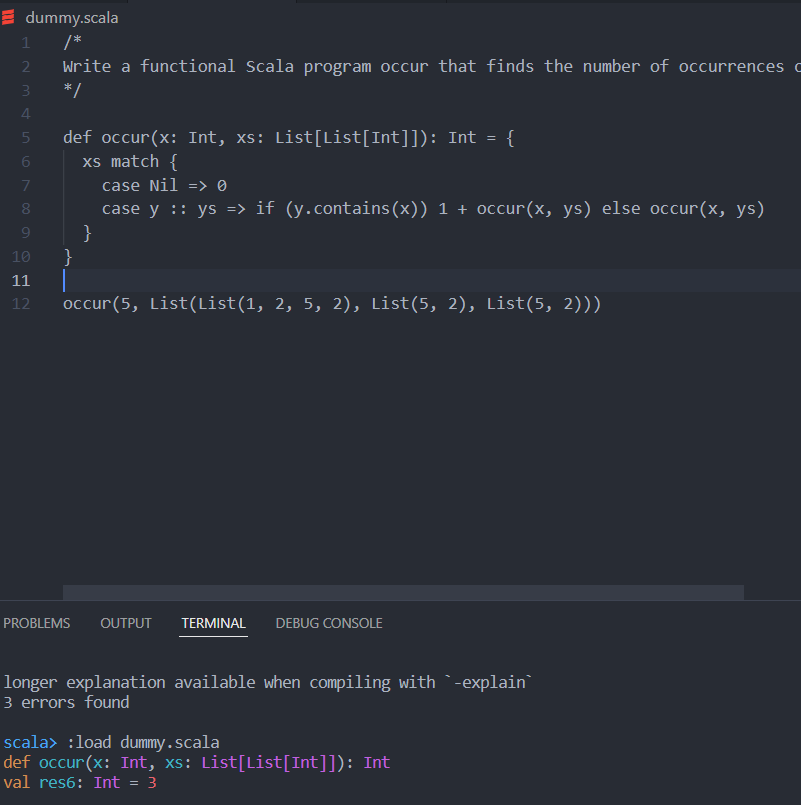
**Logic:**

**Code:**

def occur(x: Int, xs: List[List[Int]]): Int = {  
 xs match {  
 case Nil => 0  
 case y :: ys => if (y.contains(x)) 1 + occur(x, ys) else occur(x, ys)  
 }  
}  
  
occur(5, List(List(1, 2, 5, 2), List(5, 2), List(5, 2)))

**Output:**

****

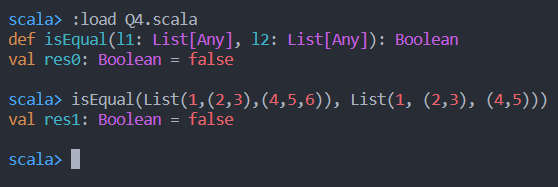
****

1. **You are aware of the == that checks the equality of two lists. Write a functional Scala program isEqual that accepts two lists (may even contain nested list elements!!) and checks whether the two lists are equal? For example isEqual (1,(2,3),(4,5,6)) (1, (2,3), (4,5)) should return False.**

**Code:**

def isEqual(l1:List[Any], l2:List[Any]):Boolean = {  
 if (l1.length != l2.length) false  
 else if (l1.isEmpty && l2.isEmpty) true  
 else if (l1.head != l2.head) false  
 else isEqual(l1.tail, l2.tail)  
}  
  
isEqual(List(1,(2,3),(4,5,6)), List(1, (2,3), (4,5)))

**Output:**

****