1.Write a class AccountInfo with methods deposit and withdraw, and a read only property balance

class AccountInfo {  
  
 private val *value*=0  
  
 def deposit() {}  
 def withdraw() {}  
  
}

2.Write an object Conversions with methods inchestoFeet, milestoKms and

poundsToKilos and invoke its methods from a class of your choice

object Conversions {  
 def inchesToCentimeters(inches: Double) = inches \* 2.54  
  
 def gallonsToLiters(gallons: Double) = gallons \* 3.78541  
  
 def milesToKilometers(miles: Double) = miles \* 1.60934  
}

class UnitConversion(val conversionFactor: Double) {  
 private def convert(value: Double) = value \* conversionFactor  
  
 def apply(value: Double) = convert(value)  
  
}  
  
object InchesToCentimeters extends UnitConversion(2.54) {}  
object GallonsToLiters extends UnitConversion(3.78541) {}  
object MilesToKilometers extends UnitConversion(1.60934) {}

3. Extend the following BankAccount class to a CheckingAccount class that charges $1 for every deposit and withdrawal

class BankAccount(initBal: Double) {  
 private var *balance* = initBal  
  
 def deposit(amount: Double) = {  
 *balance* += amount;  
 *balance* }  
  
 def withdraw(amount: Double) = {  
 *balance* -= amount;  
 *balance* }  
  
}

class CheckingAccount (initBal: Double) extends BankAccount (initBal: Double){  
 override def deposit(amount: Double) = {super.deposit(amount - 1) }  
 override def withdraw(amount: Double) = { super.withdraw(amount + 1) }  
}  
  
  
object CheckingAccountTester extends App {  
 private var *account* = new CheckingAccount(1000)  
 *account*.deposit(100)  
 *account*.withdraw(100)  
 *assert*(*account*.withdraw(997) == 0)  
}

4. Write a Scala program to get the largest element of an array using reduceLeft

object ReduceLeftExample {  
 def main(args: Array[String]): Unit = {  
 val a = *Array*(20, 12, 6, 15, 2, 9)  
 *println*("Min value " + a.reduceLeft(\_ min \_))  
 *println*("Max value " + a.reduceLeft(\_ max \_))  
 }  
}

5. Implement the factorial function using to and reduceLeft, without a loop or recursion

def factorial(n: Int): Int = if (n <= 0) 1 else (1 to n).reduceLeft(\_ \* \_)

6. Write a Scala code which reverses the lines of a file (makes the first line as the last one, and so on)

import java.io.PrintWriter  
  
class FileContentReverse {  
  
 private val *filename* = "/tmp/quote.txt"  
 io.Source.*fromFile*(*filename*)  
 .getLines.toArray  
 .reverse  
 .mkString("\n")  
  
  
}

7. Write a Scala code which reads a file and prints all words with more than 10 characters. Can you write all of it in a single line?

import scala.io.Source  
object ReadFileAndPrint {  
 def main (args: Array[String]) {  
 val filename = "filetest.txt"  
 for (line <- Source.*fromFile*(filename).getLines){  
 var a = line.split(" ")  
 for (m1 <- 0 to a.size-1) {  
 if (a(m1).length > 10)  
 *println*(a(m1))  
 }  
}