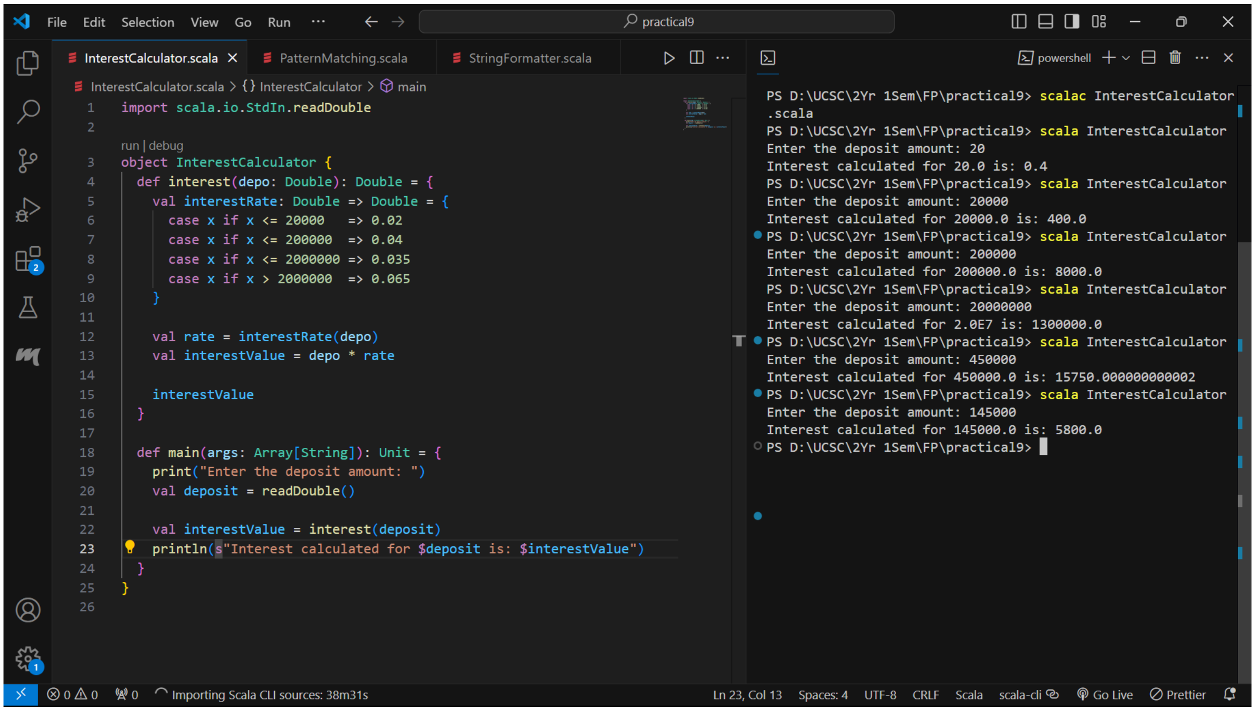
Practical 09 - 22001956

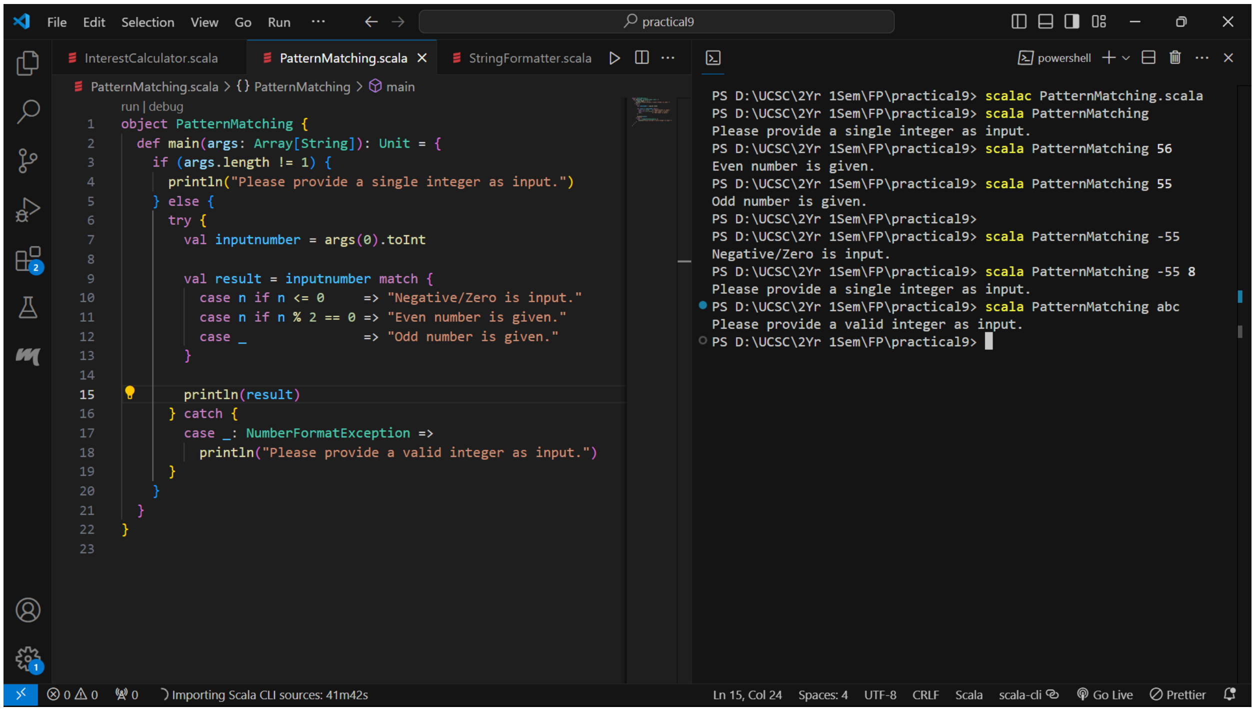
Question 01



```scala
import scala.io.StdIn.readDouble

object InterestCalculator {
  def interest(depo: Double): Double = {
    val interestRate: Double => Double = {
      case x if x <= 20000   => 0.02
      case x if x <= 200000  => 0.04
      case x if x <= 2000000 => 0.035
      case x if x > 2000000  => 0.065
    }

    val rate = interestRate(depo)
    val interestValue = depo * rate

    interestValue
  }

  def main(args: Array[String]): Unit = {
    print("Enter the deposit amount: ")
    val deposit = readDouble()

    val interestValue = interest(deposit)
    println(s"Interest calculated for $deposit is: $interestValue")
  }
}
```

```
PS D:\UCSC\2Yr 1Sem\FP\practical9> scalac InterestCalculator
.scala
PS D:\UCSC\2Yr 1Sem\FP\practical9> scala InterestCalculator
Enter the deposit amount: 20
Interest calculated for 20.0 is: 0.4
PS D:\UCSC\2Yr 1Sem\FP\practical9> scala InterestCalculator
Enter the deposit amount: 20000
Interest calculated for 20000.0 is: 400.0
PS D:\UCSC\2Yr 1Sem\FP\practical9> scala InterestCalculator
Enter the deposit amount: 200000
Interest calculated for 200000.0 is: 8000.0
PS D:\UCSC\2Yr 1Sem\FP\practical9> scala InterestCalculator
Enter the deposit amount: 20000000
Interest calculated for 2.0E7 is: 1300000.0
PS D:\UCSC\2Yr 1Sem\FP\practical9> scala InterestCalculator
Enter the deposit amount: 450000
Interest calculated for 450000.0 is: 15750.000000000002
PS D:\UCSC\2Yr 1Sem\FP\practical9> scala InterestCalculator
Enter the deposit amount: 145000
Interest calculated for 145000.0 is: 5800.0
PS D:\UCSC\2Yr 1Sem\FP\practical9>
```

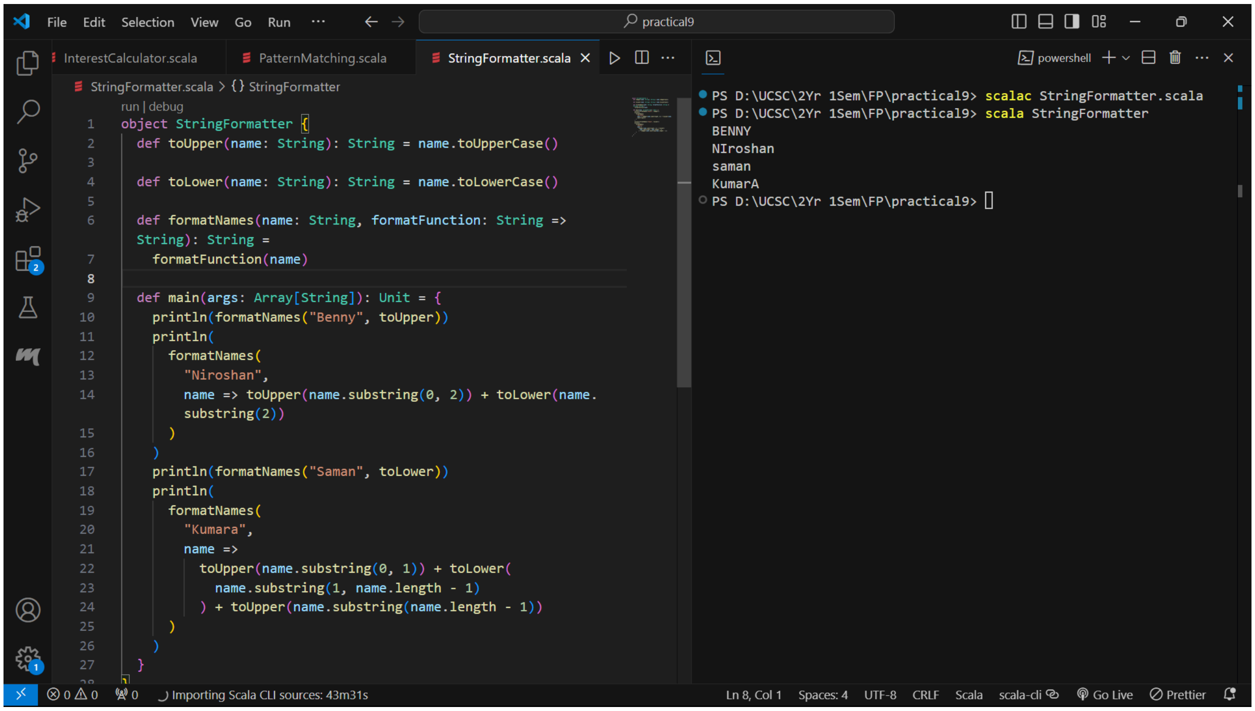
Question 02



```scala
object PatternMatching {
  def main(args: Array[String]): Unit = {
    if (args.length != 1) {
      println("Please provide a single integer as input.")
    } else {
      try {
        val inputnumber = args(0).toInt

        val result = inputnumber match {
          case n if n <= 0      => "Negative/Zero is input."
          case n if n % 2 == 0 => "Even number is given."
          case _               => "Odd number is given."
        }

        println(result)
      } catch {
        case _: NumberFormatException =>
          println("Please provide a valid integer as input.")
      }
    }
  }
}
```

```
PS D:\UCSC\2Yr 1Sem\FP\practical9> scalac PatternMatching.scala
PS D:\UCSC\2Yr 1Sem\FP\practical9> scala PatternMatching
Please provide a single integer as input.
PS D:\UCSC\2Yr 1Sem\FP\practical9> scala PatternMatching 56
Even number is given.
PS D:\UCSC\2Yr 1Sem\FP\practical9> scala PatternMatching 55
Odd number is given.
PS D:\UCSC\2Yr 1Sem\FP\practical9>
PS D:\UCSC\2Yr 1Sem\FP\practical9> scala PatternMatching -55
Negative/Zero is input.
PS D:\UCSC\2Yr 1Sem\FP\practical9> scala PatternMatching -55 8
Please provide a single integer as input.
PS D:\UCSC\2Yr 1Sem\FP\practical9> scala PatternMatching abc
Please provide a valid integer as input.
PS D:\UCSC\2Yr 1Sem\FP\practical9>
```

Question 03



```scala
object StringFormatter {
  def toUpper(name: String): String = name.toUpperCase()

  def toLower(name: String): String = name.toLowerCase()

  def formatNames(name: String, formatFunction: String => String): String =
    formatFunction(name)

  def main(args: Array[String]): Unit = {
    println(formatNames("Benny", toUpper))
    println(
      formatNames(
        "Niroshan",
        name => toUpper(name.substring(0, 2)) + toLower(name.substring(2))
      )
    )
    println(formatNames("Saman", toLower))
    println(
      formatNames(
        "Kumara",
        name =>
          toUpper(name.substring(0, 1)) + toLower(
            name.substring(1, name.length - 1)
          ) + toUpper(name.substring(name.length - 1))
      )
    )
  }
}
```

```
PS D:\UCSC\2Yr 1Sem\FP\practical9> scalac StringFormatter.scala
PS D:\UCSC\2Yr 1Sem\FP\practical9> scala StringFormatter
BENNY
NIroshan
saman
KumarA
PS D:\UCSC\2Yr 1Sem\FP\practical9>
```