

Q1:

The screenshot shows the IntelliJ IDEA interface with the file `Q1.scala` open. The code defines an object `Q1` with a `main` method that reads an integer from standard input and prints whether it is prime or not. It also contains a helper function `isPrime` that uses recursion and modulus operations to check for primality.

```
2   import scala.io.StdIn
3
4   object Q1 {
5     def main(args: Array[String]): Unit = {
6       println("Enter a integer: \n")
7       val number = StdIn.readInt()
8
9       if (isPrime(number))
10         println(s"$number is a prime number")
11       else
12         println(s" $number is not a prime number")
13     }
14
15     def isPrime(n: Int): Boolean = {
16       def divideBy(m: Int, divisor: Int): Boolean = {
17         if (divisor <= 1) false
18         else if (m % divisor == 0) true
19         else divideBy(m, divisor - 1)
20       }
21
22       if (n <= 1) false
23       else divideBy(n, Math.sqrt(n).toInt)
24     }
25   }
26 }
```

The screenshot shows a terminal window titled "Run: Q1". It displays the command "C:\Program Files\Java\jdk-20\bin\java.exe" "-javaagent:...". Below the command, the text "Enter a integer:" is displayed, indicating the program is waiting for user input.

Q2:

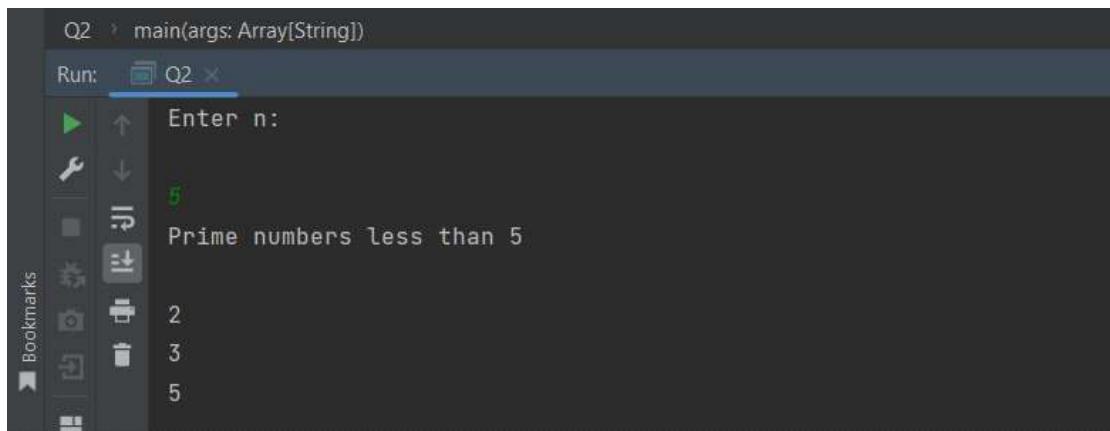
The screenshot shows the IntelliJ IDEA interface with the file `Q2.scala` open. The code defines an object `Q2` with a `main` method that prints prime numbers less than a given input. It includes a recursive helper function `primeSeq` and a helper function `isPrime` that uses a modulus operation to check for primality.

```
1   import scala.io.StdIn
2
3   object Q2 {
4     def main(args : Array[String]) : Unit = {
5       println("Enter n: \n")
6       val n = StdIn.readInt()
7       println("Prime numbers less than " + n + "\n")
8       primeSeq(n)
9     }
10
11     def primeSeq(n:Int) : Unit = {
12       if (n == 2) {
13         println(n)
14       }
15       else if (n > 2) {
16         primeSeq(n - 1)
17         if (isPrime(n)) {
18           println(n)
19         }
20       }
21     }
22
23     def isPrime(number:Int) : Boolean = {
24       def isDivisible(divisor : Int) : Boolean = {
25
26         if(divisor == 1) true
27         else if(number % divisor == 0) false
28         else isDivisible(divisor - 1)
29       }
30       isDivisible(number - 1)
31     }
32   }
33 }
```

Q2 ➔ main(args: Array[String])

Run: Q2

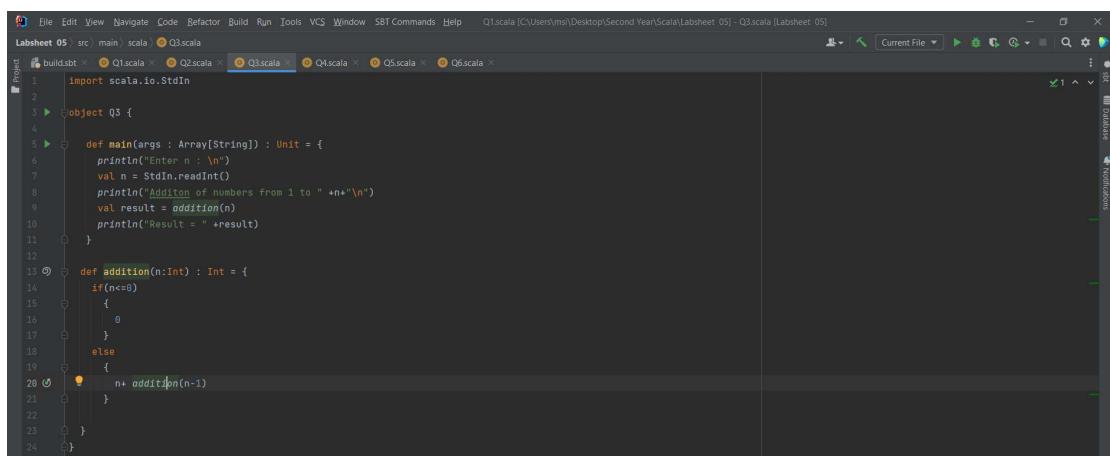
Enter n:
5
Prime numbers less than 5
2
3
5



Q3:

Labsheet 05 src/main.scala

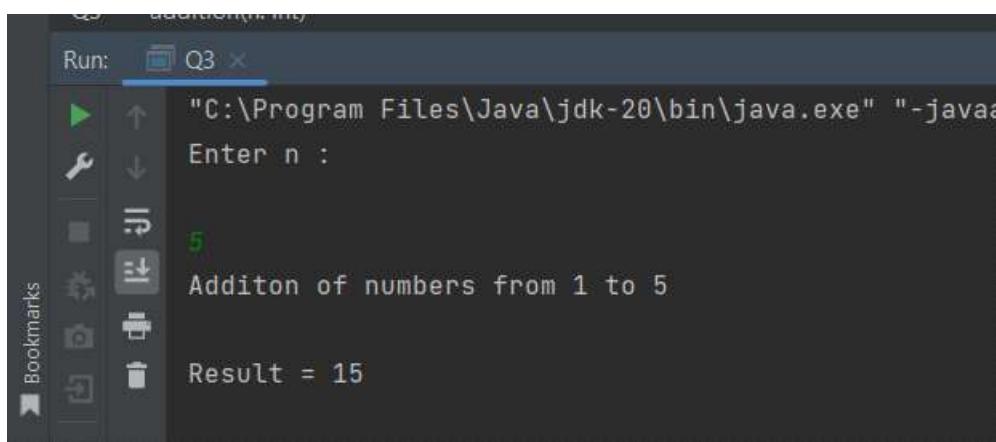
```
object Q3 {  
    def main(args : Array[String]) : Unit = {  
        println("Enter n : ")  
        val n = StdIn.readInt()  
        println("Additon of numbers from 1 to " +n+"\n")  
        val result = addition(n)  
        println("Result = " +result)  
    }  
  
    def addition(n:Int) : Int = {  
        if(n>=0)  
        {  
            0  
        }  
        else  
        {  
            n+ addition(n-1)  
        }  
    }  
}
```



Run: Q3

"C:\Program Files\Java\jdk-20\bin\java.exe" "-javaagent:D:\Programs\IntelliJ IDEA 2023.2.1\lib\idea_rt.jar" -Dfile.encoding=UTF-8

Enter n :
5
Additon of numbers from 1 to 5
Result = 15



Q4:

```
Q4.scala (C:\Users\msl\Desktop\Second Year\Scala\Labsheet 05) - Q4.scala (Labsheet 05)
File Edit View Navigate Code Refactor Build Run Tools VCS Window SBT Commands Help
1 import scala.io.StdIn
2
3 object Q4 {
4
5   def main(args : Array[String]) : Unit = {
6     println("Enter n : \n")
7     val n = StdIn.readInt()
8     println("Even numbers and odd numbers from 1 to " + n)
9     display(n)
10  }
11
12 def display(n:Int) : Unit = {
13   if(n>0)
14   {
15     display(n-1)
16     if(isEven(n))
17       println( n+ " is even")
18     else
19       println( n+ " is odd")
20   }
21 }
22
23 def isEven(number : Int) : Boolean ={
24   if(number ==0 ) true
25   else isOdd(number-1)
26 }
27
28 def isOdd(number : Int) : Boolean ={
29   if(number == 0) false
30   else isEven(number - 1)
31 }
32
```

```
Run: Q4
"C:\Program Files\Java\jdk-20\bin\java.exe" "-javaagent"
Enter n :
10
Even numbers and odd numbers from 1 to 10
1 is odd
2 is even
3 is odd
4 is even
5 is odd
6 is even
7 is odd
8 is even
9 is odd
10 is even
```

Q5:

The screenshot shows the IntelliJ IDEA interface with the following details:

- File Menu:** File, Edit, View, Navigate, Code, Refactor, Build, Run, Tools, VCS, Window, SBT Commands, Help.
- Project Bar:** LabSheet 05, build.sbt, Q1.scala, Q2.scala, Q3.scala, Q4.scala, Q5.scala (Current File), Q6.scala.
- Code Editor:** The code for `Q5.scala` is displayed. It contains a main function that reads an integer from the user, calculates the sum of even numbers up to that integer, and prints the result. It also includes a recursive helper function `addition` for calculating the sum of even numbers.
- Toolbars and Status Bar:** Standard IntelliJ IDEA toolbars and status bar.

```
import scala.io.StdIn
object Q5 {
    def main(args: Array[String]): Unit = {
        println("Enter n : ")
        val n = StdIn.readInt()
        println("Sum of all even numbers upto " + n + "\n")
        val result = addition(n)
        println("Result : " + result)
    }
    def addition(n: Int): Int = {
        if (n <= 0) {
            0
        } else if (n % 2 == 0) {
            n + addition(n - 2)
        } else {
            addition(n - 1)
        }
    }
}
```

```
Run: Q5
" C:\Program Files\Java\jdk-20\bin\java.exe" "-javaagent:D:\Program Files\Java\Agent\jolokia.jar"
Enter n :
10
Sum of all even numbers upto 10
Result : 30
```

q6:

```
File Edit View Navigate Code Refactor Build Run Tools VCS Window SBT Commands Help Q1.scala [C:\Users\msi\Desktop\Second Year\Scala\Labsheet 05] - Q6.scala [Labsheet 05]
Labsheet 05 | src | main | scala | Q6.scala
Project build.sbt < Q1.scala < Q2.scala < Q3.scala < Q4.scala < Q5.scala < Q6.scala
1 import scala.io.StdIn
2
3 object Q6 {
4
5   def main(args : Array[String]) : Unit = {
6     println("Enter n: \n")
7     val n = StdIn.readInt()
8     sequence(n)
9   }
10
11   def fibonacci(n:Int) : Int = n match {
12     case 0 => 0
13     case x if x == 1 => 1
14     case _ => fibonacci(n - 1) + fibonacci(n - 2)
15   }
16
17   def sequence(n: Int) : Unit = {
18     if(n>=0) sequence(n-1)
19     println(fibonacci(n))
20   }
21
22
23
24 }
```

