

SCALA PROGRAMMING

TAIL RECURSION

Code-1:

```
import scala.annotation.tailrec

// 1 - basic recursive factorial method
def factorial(n: Int): Int = if (n == 0) 1 else n * factorial(n-1)

// 2 - tail-recursive factorial method
def factorial2(n: Long): Long = {
  @tailrec
  def factorialAccumulator(acc: Long, n: Long): Long = {
    if (n == 0) acc else factorialAccumulator(n*acc, n-1)
  }
  println(factorialAccumulator(1, 5))
  println(factorial(6))
}
```

Output-1:

Output:

120
720

Screenshot-1:

```
1 import scala.annotation.tailrec
2 // 1 - basic recursive factorial method
3 def factorial(n: Int): Int = if (n == 0) 1 else n * factorial(n-1)
4 // 2 - tail-recursive factorial method def factorial2(n: Long): Long = {
5   @tailrec
6   def factorialAccumulator(acc: Long, n: Long): Long = {
7     if (n == 0) acc else factorialAccumulator(n*acc, n-1)
8   }
9   println(factorialAccumulator(1, 5))
10
11   println(factorial(6))
12
13 }
```

STDIN

Input for the program (Optional)

Output:

120
720

Code-2:

```
import scala.annotation.tailrec

// 1 - basic recursive factorial method
def factorial(n: Int): Int = if (n == 0) 1 else n * factorial(n-1)

// 2 - tail-recursive factorial method
def factorial2(n: Long): Unit = {
  @tailrec
  def factorialAccumulator(acc: Long, n: Long): Long = {
    if (n == 0) acc else factorialAccumulator(n*acc, n-1)
  }
  println(factorialAccumulator(1, n))
}

factorial2(3)
println(factorial(5))
```

Output-2:

Output:

6
120

Screenshot-2:

```
1 import scala.annotation.tailrec
2 // 1 - basic recursive factorial method
3 def factorial(n: Int): Int = if (n == 0) 1 else n * factorial(n-1)
4 // 2 - tail-recursive factorial method
5 def factorial2(n: Long): Unit = {
6   @tailrec
7   def factorialAccumulator(acc: Long, n: Long): Long = {
8     if (n == 0) acc else factorialAccumulator(n*acc, n-1)
9   }
10  println(factorialAccumulator(1, n))
11 }
12 factorial2(3)
13 println(factorial(5))
14
```

STDIN

Input for the program (Optional)

Output:

6
120

Taking input from the user:

Code-3:

```
import scala.annotation.tailrec
import scala.io.StdIn

// 1 - basic recursive factorial method
def factorial(n: Int): Int = if (n == 0) 1 else n * factorial(n-1)

// 2 - tail-recursive factorial method
def factorial2(n: Long): Unit = {
  @tailrec
  def factorialAccumulator(acc: Long, n: Long): Long = {
    if (n == 0) acc else factorialAccumulator(n*acc, n-1)
  }
  println(factorialAccumulator(1, n))
}

var x = StdIn.readInt()
factorial2(x)
println(factorial(5))
```

Output-3:

STDIN

3

Output:

6

120

Screenshot-3:

```
1 import scala.annotation.tailrec
2 import scala.io.StdIn
3 // 1 - basic recursive factorial method
4 def factorial(n: Int): Int = if (n == 0) 1 else n * factorial(n-1)
5 // 2 - tail-recursive factorial method
6 def factorial2(n: Long): Unit = {
7   @tailrec
8   def factorialAccumulator(acc: Long, n: Long): Long = {
9     if (n == 0) acc else factorialAccumulator(n*acc, n-1)
10  }
11  println(factorialAccumulator(1, n))
12 }
13
14 var x = StdIn.readInt()
15 factorial2(x)
16 println(factorial(5))
17
```

STDIN

3

Output:

6

120