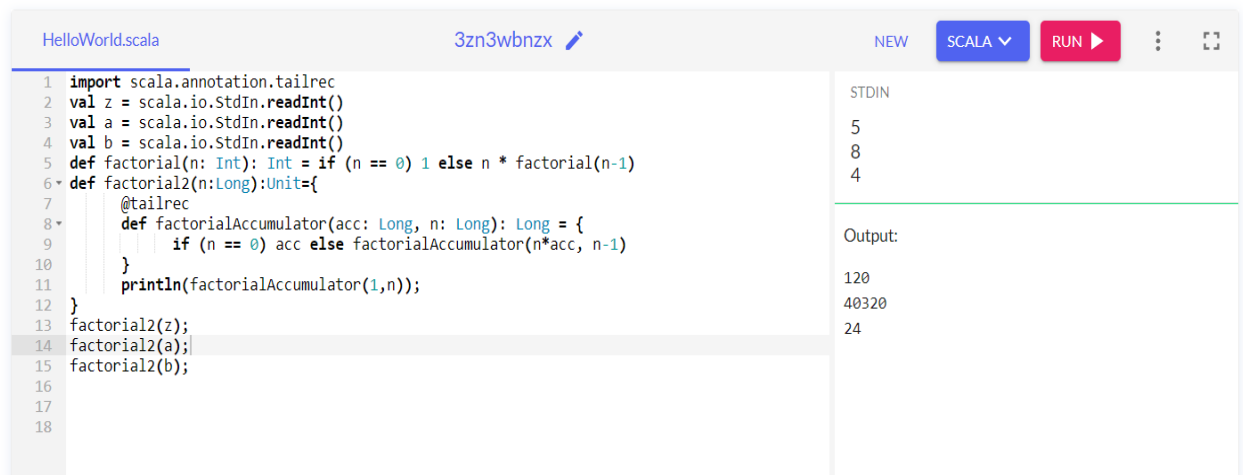


NAME – TAPABRATA BANERJEE

USN- 22BTRAD030

SCALA PROGRAMMING

Q. Tail Recursion:



The screenshot shows a Scala IDE with a file named `HelloWorld.scala`. The code defines a tail-recursive factorial function using an accumulator. The function `factorial2` takes a `Long` and returns a `Unit`. It uses `@tailrec` to ensure tail recursion. The `factorialAccumulator` helper function takes an accumulator and a number, and returns a `Long`. The main function `factorial2` calls `factorialAccumulator` with an initial value of 1 and the input `n`. The IDE shows the execution results for three inputs: 5, 8, and 4. The output for 5 is 120, for 8 is 40320, and for 4 is 24.

```
1 import scala.annotation.tailrec
2 val z = scala.io.StdIn.readInt()
3 val a = scala.io.StdIn.readInt()
4 val b = scala.io.StdIn.readInt()
5 def factorial(n: Int): Int = if (n == 0) 1 else n * factorial(n-1)
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 factorial2(z);
14 factorial2(a);
15 factorial2(b);
16
17
18
```

STDIN

5
8
4

Output:

120
40320
24