Anuvind M P
AM.U4AIE22010

1.

```
object SumTriple {
    def sumTriple(a: Int, b: Int): Int = {
        if (a == b) (a + b) * 3 else a + b
    }
    def main(args: Array[String]): Unit = {
        println(sumTriple(12, 6))
        println(sumTriple(6, 3))
    }
}
```

Output:

```
18
9
```

```
object CheckNumbers {
    def check(a: Int, b: Int): Boolean = {
        a == 22 || b == 22 || (a + b == 32)
    }
    def main(args: Array[String]): Unit = {
        println(check(18, 9))val prefix = str.take(5)
        prefix + str + prefix
        println(check(7, 14))
        println(check(10, 22))
        println(check(5, 10))
}
}
```

Output:

```
false
false
true
false
```

2.

```
object RemoveChar {
    def removeChar(str: String, pos: Int): String = {
        if (pos >= 0 && pos < str.length) str.take(pos) + str.drop(pos + 1)
else
        str
    }
    def main(args: Array[String]): Unit = {
        println(removeChar("Anuvind", 2))
        println(removeChar("hehehe", 3))
    }
}
```

Output:

```
Anvind
hehhe
```

```
object ModifyString {
    def modifyString(str: String): String = {
        val prefix = str.take(5)
        prefix + str + prefix
    }
    def main(args: Array[String]): Unit = {
        println(modifyString("anuvindmp"))
    }
}
```

Output:

```
anuvianuvindmpanuvi
```

3.

```
object MultiplicationTable {
    def printTable(n: Int): Unit = {
        for (i <- 1 to 10) {
        println(s"$n x $i = ${n * i}")
    }
}
    def main(args: Array[String]): Unit = {
        printTable(5)
}
}
```

Output:

```
5 x 1 = 5
5 x 2 = 10
5 x 3 = 15
5 x 4 = 20
5 x 5 = 25
5 x 6 = 30
5 x 7 = 35
5 x 8 = 40
5 x 9 = 45
5 x 10 = 50
```

```
object LargestElement {
    def findLargest(arr: Array[Int]): Int = {
        arr match {
        case Array() => throw new IllegalArgumentException("Array is empty")
        case _ => arr.max
    }
}
    def main(args: Array[String]): Unit = {
        val numbers = Array(7,45,223,76,67)
        println(findLargest(numbers))
    }
}
```

Output:

```
223
```

4.

```
object ProductOfDigits {
    def productOfDigits(n: Int): Int = {
        n.toString.map(_.asDigit).product
    }
    def main(args: Array[String]): Unit = {
        println(productOfDigits(852))
    }
}
```

Output:

```
80
```

```
object PerfectSquare {
    def isPerfectSquare(n: Int): Boolean = {
        val sqrt = Math.sqrt(n).toInt
        sqrt * sqrt == n
    }
    def main(args: Array[String]): Unit = {
        println(isPerfectSquare(100))
        println(isPerfectSquare(13))
    }
}
```

Output:

```
true
false
```