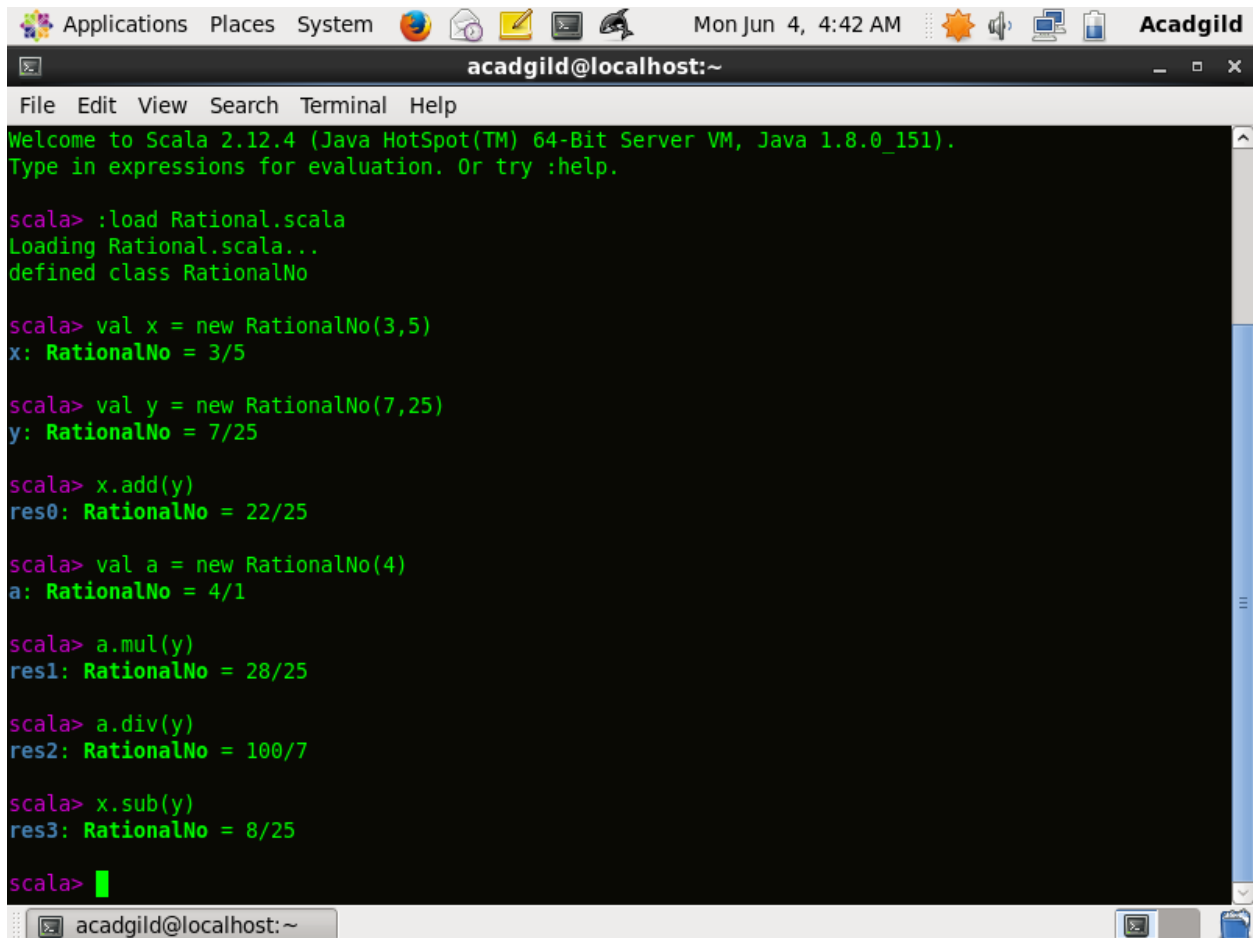


The file in which my program has name Rational.scala

So Using the command :Load Rational.scala

The name of the class is RationalNo

The function names are add, sub, mul, div



The screenshot shows a terminal window titled "acadgild@localhost:~" with a menu bar (File, Edit, View, Search, Terminal, Help) and a system bar at the top (Applications, Places, System, Mon Jun 4, 4:42 AM, Acadgild). The terminal output is as follows:

```
Welcome to Scala 2.12.4 (Java HotSpot(TM) 64-Bit Server VM, Java 1.8.0_151).
Type in expressions for evaluation. Or try :help.

scala> :load Rational.scala
Loading Rational.scala...
defined class RationalNo

scala> val x = new RationalNo(3,5)
x: RationalNo = 3/5

scala> val y = new RationalNo(7,25)
y: RationalNo = 7/25

scala> x.add(y)
res0: RationalNo = 22/25

scala> val a = new RationalNo(4)
a: RationalNo = 4/1

scala> a.mul(y)
res1: RationalNo = 28/25

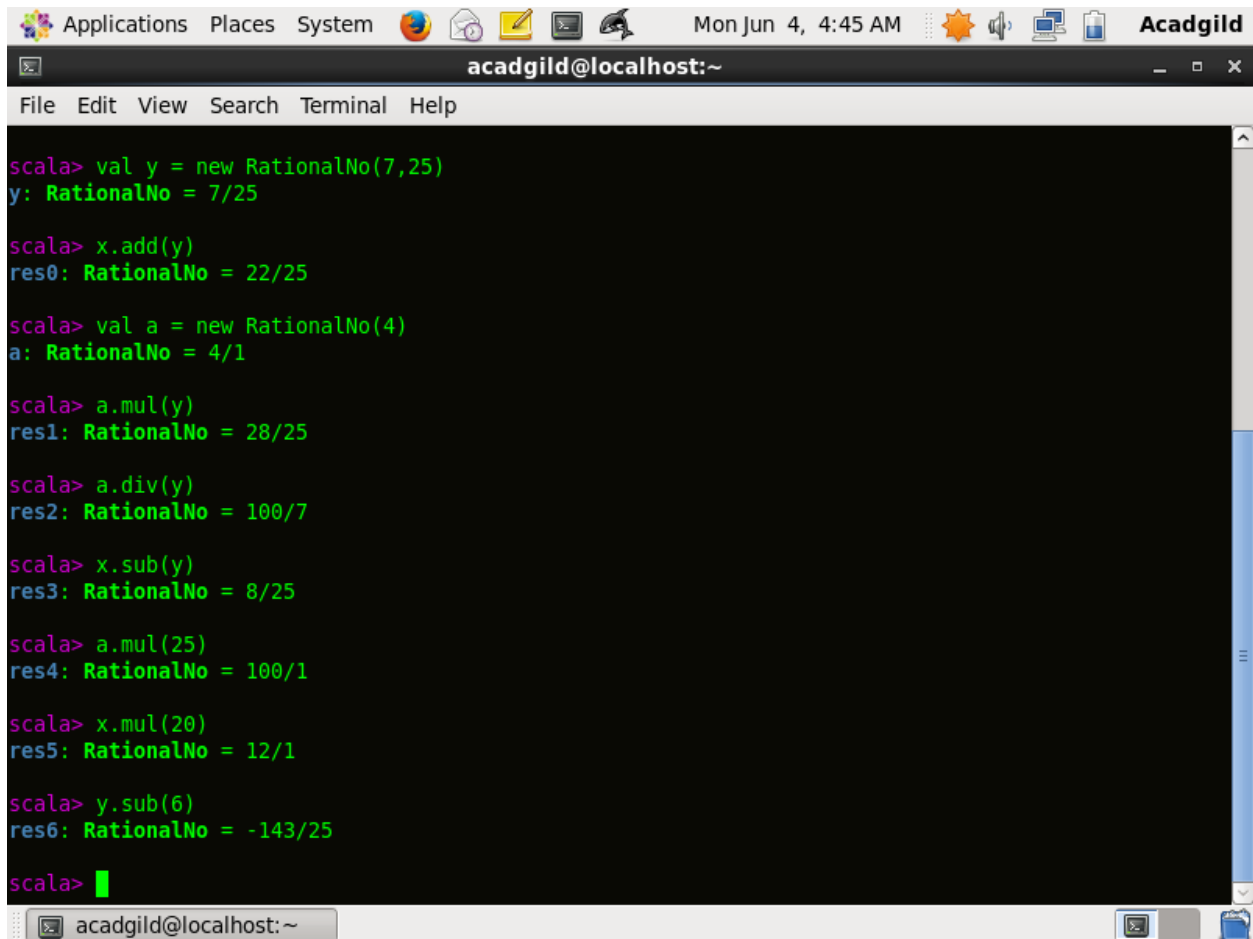
scala> a.div(y)
res2: RationalNo = 100/7

scala> x.sub(y)
res3: RationalNo = 8/25

scala> 
```

I have also passed integer parameters with the help of function overloading

Val a is created just by passing an integer with the help of auxillary constructor



The screenshot shows a Scala REPL window titled 'acadgild@localhost:~'. The window has a menu bar with 'File', 'Edit', 'View', 'Search', 'Terminal', and 'Help'. The main area displays the following Scala code and its output:

```
scala> val y = new RationalNo(7,25)
y: RationalNo = 7/25

scala> x.add(y)
res0: RationalNo = 22/25

scala> val a = new RationalNo(4)
a: RationalNo = 4/1

scala> a.mul(y)
res1: RationalNo = 28/25

scala> a.div(y)
res2: RationalNo = 100/7

scala> x.sub(y)
res3: RationalNo = 8/25

scala> a.mul(25)
res4: RationalNo = 100/1

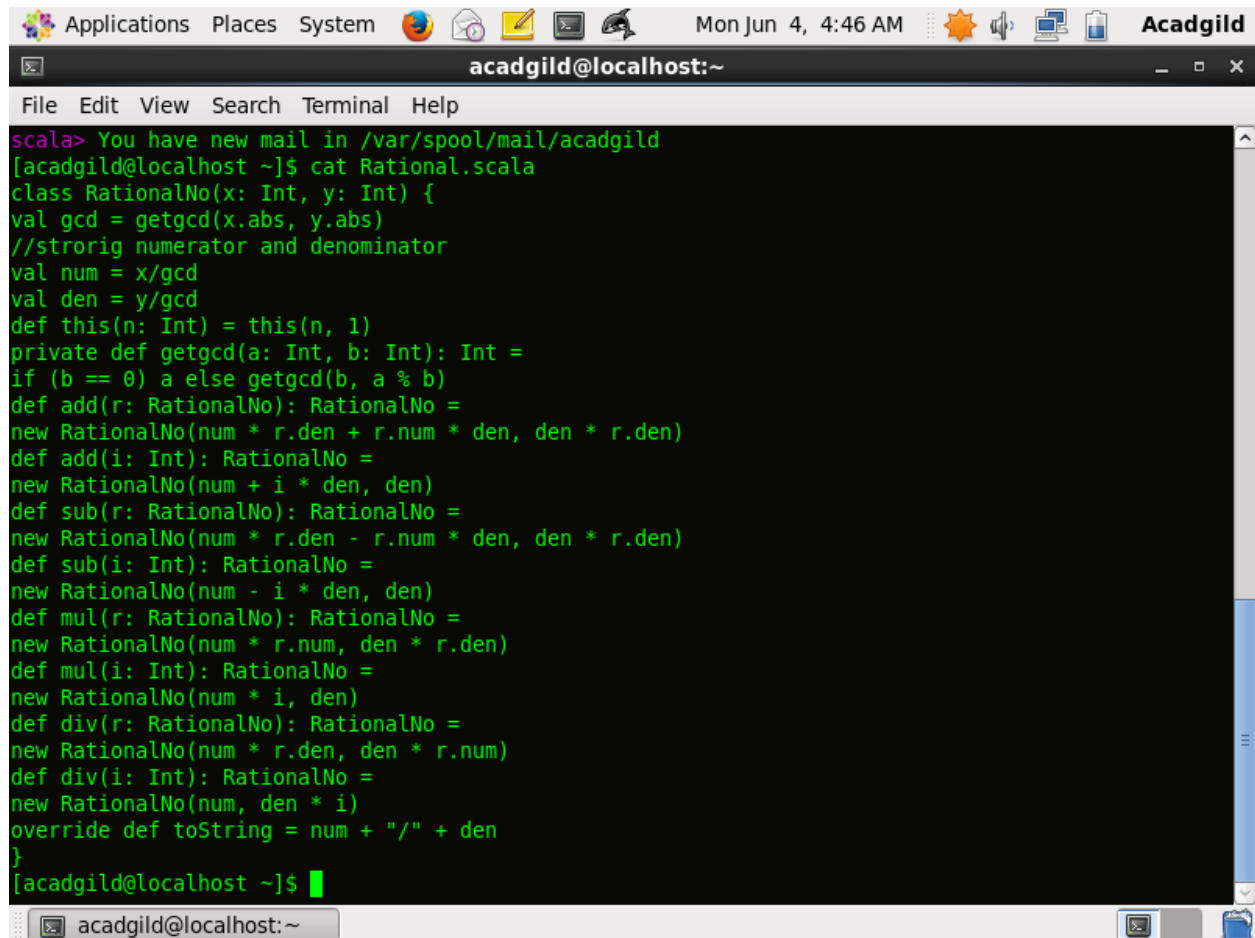
scala> x.mul(20)
res5: RationalNo = 12/1

scala> y.sub(6)
res6: RationalNo = -143/25

scala>
```

The window's title bar includes system icons for applications, places, system, and network, along with the date and time 'Mon Jun 4, 4:45 AM'. The bottom status bar shows the user 'acadgild@localhost:~' and a terminal icon.

Code



The screenshot shows a Linux desktop environment. At the top is a panel with icons for Applications, Places, and System, followed by the date and time 'Mon Jun 4, 4:46 AM', and the username 'Acadgild'. Below this is a terminal window titled 'acadgild@localhost:~'. The terminal has a menu bar with 'File', 'Edit', 'View', 'Search', 'Terminal', and 'Help'. The terminal content shows a Scala prompt 'scala>' followed by a message about new mail, then a shell prompt '[acadgild@localhost ~]\$' followed by the command 'cat Rational.scala'. The output is the source code for a Scala class 'RationalNo' which implements rational number arithmetic. The code includes methods for gcd, addition, subtraction, multiplication, and division, both as functions and as infix operators. The terminal ends with the shell prompt '[acadgild@localhost ~]\$' and a cursor.

```
scala> You have new mail in /var/spool/mail/acadgild
[acadgild@localhost ~]$ cat Rational.scala
class RationalNo(x: Int, y: Int) {
  val gcd = getgcd(x.abs, y.abs)
  //strorig numerator and denominator
  val num = x/gcd
  val den = y/gcd
  def this(n: Int) = this(n, 1)
  private def getgcd(a: Int, b: Int): Int =
    if (b == 0) a else getgcd(b, a % b)
  def add(r: RationalNo): RationalNo =
    new RationalNo(num * r.den + r.num * den, den * r.den)
  def add(i: Int): RationalNo =
    new RationalNo(num + i * den, den)
  def sub(r: RationalNo): RationalNo =
    new RationalNo(num * r.den - r.num * den, den * r.den)
  def sub(i: Int): RationalNo =
    new RationalNo(num - i * den, den)
  def mul(r: RationalNo): RationalNo =
    new RationalNo(num * r.num, den * r.den)
  def mul(i: Int): RationalNo =
    new RationalNo(num * i, den)
  def div(r: RationalNo): RationalNo =
    new RationalNo(num * r.den, den * r.num)
  def div(i: Int): RationalNo =
    new RationalNo(num, den * i)
  override def toString = num + "/" + den
}
[acadgild@localhost ~]$
```