Mathematics

To begin:

- In Eclipse create a Scala project called MathLab.
- Add a worksheet to this project called session
- Add a Scala interpreter to this project (just for fun)

In the session worksheet define and test the following functions. Your implementations should be as concise as possible. You should use library functions whenever possible. You should use operator symbols whenever possible.

Hints:

- Scala numeric classes include: Byte, Short, Int, Long, Float, Double
- Unlike Java, Scala doesn't have scalar types such as int and double. Remember, everything in Scala is an object!
- Of course the Numeric classes have useful methods, but more useful methods can be found in the scala.math package:

  http://www.scala-lang.org/api/current/#scala.math.package

- Import all definitions from scala.math. Recall that this only means you don't have to call math.sin(x), you can call sin(x) instead. To do this, begin your session with

```
import scala.math._
```

To end:

- Export session.sc to your file system and send it to the grader by the deadline.

1. Problem

Write a function that given the coefficients of a quadratic polynomial $ax^2 + bx + c$, returns the real roots:

```
solve(2, -2, -4)      > res0: Option[(Double, Double)] = Some((2.0,-1.0))
solve(1, 0, 1)        > res1: Option[(Double, Double)] = None
solve(1, 0, -1)       > res2: Option[(Double, Double)] = Some((1.0,-1.0))
```

Notes:

- Our version of solve returns an option containing a tuple. Options and tuples are discussed here: Options, Maps, and Tuples.
- When there are two real roots, as in the first example, solve returns a Some object containing the roots as a pair of doubles.
- When there are no roots, as in the second example, solve returns None.
- How would you modify solve to return complex roots

2. Problem

Write a function that computes the distance between two points in the x-y plane:

```
dist((1, 1), (0, 0)) = 1.4142135623730951  // = sqrt(2)
dist((3, 0), (0, 0)) = 3.0
```

Notes:

- Code up the distance formula, don't use a pre-defined distance function if there is one.
- Notice that the inputs of dist are pairs. See the earlier discussion for more information on how to dissect them.
- Add a comment to your session in which you write out the exact type of dist. Do you understand the notation?
- How would you modify dist to compute the distance between points in 3 dimensional space? in N dimensional space?

3. Problem

Write a function that computes the dot product of two three dimensional vectors:

```
dot((2.0, 3, 4), (2, 2.0, 2)) = 18.0
```

Notice that vectors are being represented as triples

4. Problem

Write a function that computes the gravitational force in Newtons between two objects given their masses in kilograms and the distance between them in meters:

```
force(m1, m2, d)
```

How would you modify force if d was miles instead of meters?

5. Problem

Write functions that compute the mean and standard deviation of an array of numbers:

```
mean(Array(2.0, 3, 4, 5)) = 3.5
stdDev(Array(2, 3.0, 4, 5)) =
```

6. Problem

Write a function that determines if an integer input is prime:

```
isPrime(0) = false
isPrime(1) = false
isPrime(2) = true
isPrime(8) = false
etc.
```

Notes:

- isPrime should throw an exception if its input is negative.
- Write isPrime from scratch, don't use a predefined function, if there is one.
- How would you write a twin prime detector? Examples: (3, 5), (11, 13), (17, 19)

7. Problem

Implement Euler's phi function:

```
phi(n) = # of integers k such that 0 < k <= n and gcd(n, k) = 1
```

For example:

```
phi(9) = 6
phi(10) = 4
```

8. Problem

Implement a function called rollDice:

```
rollDice => (3, 2)
rollDice => (1, 6)
rollDice => (3, 3)
```

Hints:

- scala.util.Random is Scala's random number generator. It's a pre-defined object, though, not a class.
- How would you implement rollDice using the function math.random?
- Will the same sequence of rolls repeat if you restart the interpreter? How can this be prevented?