Problem 1

Create a class **OneLiners** with four static functions described below. Try to implement them as 'one-liners', i.e. in such a way that each of them consists only of one **return** statement.

- `public static boolean hasTwoRoots(double a, double b, double c)` returns **true** if, and only if, the equation ax^2+bx+c has two different solutions, and **false** otherwise;
- `public static boolean monot(double a, double b, double c)` returns **true** if, and only if, the numbers `a`, `b` and `c` constitute a strictly increasing sequence (each subsequent term is bigger than the previous) or a strictly decreasing one, and **false** otherwise;
- `public static double maxEl(double a, double b, double c))` returns the maximum value of the numbers `a`, `b` and `c`;
- `public static int numPos(double a, double b, double c)` returns the number of positive values among `a`, `b` and `c`.

For example, the following program,

download *OneLiners.java*

```
public class OneLiners {
    public static boolean hasTwoRoots(double a,
                                      double b, double c) {
        // ...
    }
    public static boolean monot(double a,
                                double b, double c) {
        // ...
    }
    public static double maxEl(double a,
                               double b, double c) {
        // ...
    }
    public static int numPos(double a,
                             double b, double c) {
        // ...
    }

    public static void main(String[] args) {
        double a = 2, b = 3, c = 1;
        System.out.println("(a, b, c,)=(" + a + ", " +
                b + ", " + c + "): hasTwoRoots? " +
                hasTwoRoots(a,b,c));
```

```
        a = 0; b = 2; c = 1;
        System.out.println("(a, b, c,)=(" + a + ", " +
                b + ", " + c + "): hasTwoRoots? " +
                hasTwoRoots(a,b,c));

        a = 2; b = 1; c = -1;
        System.out.println("(a, b, c,)=(" + a + ", " +
                b + ", " + c + "): monot? " +
                monot(a,b,c));
        a = 2; b = 1; c = 2;
        System.out.println("(a, b, c,)=(" + a + ", " +
                b + ", " + c + "): monot? " +
                monot(a,b,c));

        a = 2; b = 1; c = 2;
        System.out.println("(a, b, c,)=(" + a + ", " +
                b + ", " + c + "): maxEl = " +
                maxEl(a,b,c));
        a = 2; b = 2; c = 4;
        System.out.println("(a, b, c,)=(" + a + ", " +
                b + ", " + c + "): maxEl = " +
                maxEl(a,b,c));

        a = -2; b = 1; c = -3;
        System.out.println("(a, b, c,)=(" + a + ", " +
                b + ", " + c + "): numPos = " +
                numPos(a,b,c));
        a = -2; b = 1; c = 3;
        System.out.println("(a, b, c,)=(" + a + ", " +
                b + ", " + c + "): numPos = " +
                numPos(a,b,c));
    }
}
```

after implementing the functions, should print

```
(a, b, c,)=(2.0, 3.0, 1.0): hasTwoRoots? true
(a, b, c,)=(0.0, 2.0, 1.0): hasTwoRoots? false
(a, b, c,)=(2.0, 1.0, -1.0): monot? true
(a, b, c,)=(2.0, 1.0, 2.0): monot? false
(a, b, c,)=(2.0, 1.0, 2.0): maxEl = 2.0
(a, b, c,)=(2.0, 2.0, 4.0): maxEl = 4.0
(a, b, c,)=(-2.0, 1.0, -3.0): numPos = 1
(a, b, c,)=(-2.0, 1.0, 3.0): numPos = 2
```

Problem 2

Create a class containing static functions:

- `static int maxOfThree(int a, int b, int c)` returning the value of the greatest of three numbers passed to the function;
- `static int greatestDivisor(int n)` returning the greatest divisor of `n` smaller than `n` (this will be 1 for prime numbers);
- `static boolean areRelativelyPrime(int a, int b)` returning **true** if and only if `a` and `b` are relatively prime;
- `static boolean isPerfect(int n)` returning **true** if and only if `n` is a perfect number, i.e., is the sum of all its divisors (including 1 but excluding `n` itself).

In the **main** function test all these functions. Do not use any classes from packages other than basic ***java.lang*** (in particular, no collections are allowed). You can assume that the functions will be invoked with positive arguments.

For example, the following program

download *StatFuns.java*

```
public class StatFuns {
    static int maxOfThree(int a, int b, int c) {
        // ...
    }
    static int greatestDivisor(int n) {
        // ...
    }
    static boolean areRelativelyPrime(int a, int b) {
        // ...
    }
    static boolean isPerfect(int n) {
        // ...
    }
    public static void main(String[] args) {
        int x = 2, y = 3, z = 1;
        System.out.println("Max of " + x + ", " + y + ", " +
                z + " is " + maxOfThree(x, y, z));

        for (int a = 12; a < 16; ++a)
                System.out.println("Greatest divisor of " +
                    a + " is " + greatestDivisor(a));

        for (int m = 11, n = 5; m >= 3; m -= 2, n += 2)
            if (areRelativelyPrime(m, n))
                System.out.println(m + " and " + n +
                    " are relatively prime");

        for (int m = 2; m <= 100; ++m)
            if (isPerfect(m))
                System.out.println(m + " is perfect");
    }
}
```

should print

```
Max of 2, 3, 1 is 3
Greatest divisor of 12 is 6
Greatest divisor of 13 is 1
Greatest divisor of 14 is 7
Greatest divisor of 15 is 5
11 and 5 are relatively prime
9 and 7 are relatively prime
7 and 9 are relatively prime
5 and 11 are relatively prime
3 and 13 are relatively prime
6 is perfect
28 is perfect
```
