1 Expressions, Precedence, Assignments etc.

For this exercise we prepared some code here: http://lec.inf.ethz.ch/baug/informatik2/2015/ex/ex02/Precedence/Main.java. Your first task is to replace all question marks "?" in the file by a literal (a number) such that the comparison in each case yields true. E.g. in the case of

```
boolean res1 = false;
int b = 5;
int a = b = 3;
res1 = (b == ?);
```

you should replace "?" by 3 in order to get the correct answer.

Executing main should print `true` for each partial result. When finished, submit your file here: https://judge.inf.ethz.ch/team/websubmit.php?cid=60&problem=IB1502

2 Tossing a fair Dice

Download http://lec.inf.ethz.ch/baug/informatik2/2015/ex/ex02/Dice/Main.java. It comprises the linear congruential generator presented during the lecture. Your task is to complete the method

```
public static int Dice()
```

such that it returns a number between 1 and 6 using the following guidelines:

1. Draw a uniform random number using the `Uniform()` method from class LCG
2. `LCG.Uniform()` delivers a number in the interval $[0, 1)$
3. Map the floating point numbers to integer numbers 1 to 6 such that numbers in $[0, \frac{1}{6})$ are mapped to 1, numbers in $[\frac{1}{6}, \frac{1}{3})$ are mapped to 2 and so on.

Test your function using the main routine within the class. When finished, submit the file containing a correct Dice implementation here: https://judge.inf.ethz.ch/team/websubmit.php?cid=60&problem=IB1503

3 Creating Gaussian Pseudo Random Numbers

For the final part we are going to use the Box Muller transform to convert uniformly distributed random numbers, such as generated by our linear congruential generator, into Gaussian random numbers.

The Box Mueller transform converts a pair of two independent Uniformly distributed random variables U_1 and U_2 into a pair of two independent Gaussian variables Z_0 and Z_1 using the formulas

$$\begin{aligned} Z_0 &= \sqrt{-2 \ln U_1} \cos(2\pi\ U_2) \\ Z_1 &= \sqrt{-2 \ln U_1} \sin(2\pi\ U_2). \end{aligned} \tag{1}$$

Download the code skeleton http://lec.inf.ethz.ch/baug/informatik2/2015/ex/ex02/Gaussian/Main.java and complete the implementation of the method

```
public static double generateGaussian()
```

such that each call returns a new value drawn from a standard normal distribution. Implement the code adhering to the following guidelines:

1. Following the equations from above, Gaussian random variables do only come in pairs and require a pair of uniform random numbers. However, random number should not be wasted!
2. Therefore your method should calculate Z_0 and Z_1 on each odd calls (first time, third time, 5th etc.) - then only return Z_0, but store Z_1 (in a static variable).
3. Every even (2nd, 4th, 6th etc.) call it then simply returns the stored Z_1 without creating new random values.

When finished submit your implementation to the judge: https://judge.inf.ethz.ch/team/websubmit.php?cid=60&problem=IB1504