In Zeile 3 der Datei `Longtst.java` haben wir also eine Zahl verwendet, die zu groß ist. Zu groß deshalb, weil sie ja standardmäßig als **`int`**-Wert angenommen wird, aber laut Tabelle 4.2 deutlich größer als `2147483647` ist und somit nicht mehr mit 32 Bits dargestellt werden kann. Eine derartige Zahl muss explizit als längere Zahl gekennzeichnet werden! Wir ändern die Zeile deshalb wie folgt:

```
System.out.println(9223372036854775807L);
```

Durch die Hinzunahme der Endung `L` wird die Zahl als eine **`long`**-Zahl betrachtet und somit mit 64 Bits codiert (in die sie laut Tabelle gerade noch hineinpasst). Der entsprechende Datentyp heißt in Java **`long`**.
Achtung: Neben der rein dezimalen Schreibweise von ganzzahligen Werten können Literalkonstanten in Java auch als oktale Zahlen (Zahlen im Achter-System) oder als hexadezimale Zahlen (Zahlen im Sechzehner-System) geschrieben werden. Oktale Zahlen müssen mit einer führenden `0` beginnen und dürfen nur Ziffern im Bereich 0 bis 7 enthalten. Hexadezimale Zahlen müssen mit `0x` beginnen und dürfen Ziffern im Bereich 0 bis 9 und A bis F enthalten. Die drei **`int`**-Konstanten `27`, `033` und `0x1B` sind somit alternative Schreibweisen für den ganzzahligen dezimalen Wert 27.

4.3.2 Gleitkommatypen

Wir wollen eine einfache Rechung durchführen. Das folgende Programm soll das Ergebnis von `1/10` ausgeben und verwendet dazu den Divisionsoperator in Java:

```
1 public class Intdiv {
2   public static void main (String[] args) {
3     System.out.println(1/10);
4   }
5 }
```

Wir übersetzen das Programm und führen es aus. Zu unserer Überraschung erhalten wir jedoch ein vermeintlich falsches Ergebnis – und zwar die Null! Was ist geschehen?
Um zu begreifen, was eigentlich passiert ist, müssen wir uns eines klar machen: wir haben mit ganzzahligen Datentypen gearbeitet. Der Divisionsoperator ist in Java jedoch so definiert, dass die Division zweier ganzer Zahlen wiederum eine ganze Zahl (nämlich den ganzzahligen Anteil des Quotienten) ergibt. Wir erinnern uns an die Grundschulzeit – hier hätte `1/10` ebenfalls 0 ergeben – mit Rest 1. Diesen Rest können wir in Java mit dem `%`-Zeichen bestimmen. Wir ändern unser Programm entsprechend:

```
1 public class Intdiv {
2   public static void main (String[] args) {
3     System.out.print("1/10 betraegt ");
4     System.out.print(1/10);  // ganzzahliger Anteil
5     System.out.print(" mit Rest ");
6     System.out.print(1%10);  // Rest
7   }
8 }
```