Worksheet: Integer Division for Negative Numbers

1. We will examine the use of bit shifting to perform division, and compare the results for integer division in Python and Java.

   a) Convert the denary value `+14` into an 8-bit two's complement number.

   b) Perform the arithmetic operation `(+14 >> 2)` using the binary value found in part (a), then convert the value back to denary.

   c) Write the bits that are shifted out by the arithmetic operation `(+14 >> 2)`, then convert that value to denary.

   d) Convert the denary value `-14` into an 8-bit two's complement number.

   e) Perform the arithmetic operation `(-14 >> 2)` using the binary value found in part (d), then convert the value back to denary.

   f) Write the bits that are shifted out by the arithmetic operation `(-14 >> 2)`, then convert that value to denary.

   When bit shifting is used to perform integer division, the resulting value is the next integer value that is lower (more negative, not of smaller magnitude) than the real value that would be calculated by real-number division. In mathematics and computer science, this is also called the ***floor*** of the number. The floor is defined as the greatest integer less than or equal to x, and is denoted as $\lfloor x \rfloor$. Mathematically, we can write the definition as:

$$\lfloor x \rfloor = \max\{ n \in Z \mid n \leq x \}$$

Worksheet: Integer Division for Negative Numbers

2. Consider the following Java code and the resulting output.

	Java Code	Output

i)	`System.out.println(14 / 4);`
	`System.out.println(14 % 4);`
	`System.out.println(14 >> 2);`
		`3`
		`2`
		`3`

ii)	`System.out.println((-14) / 4);`
	`System.out.println((-14) % 4);`
	`System.out.println((-14) >> 2);`
		`-3`
		`-2`
		`-4`

Java integer division will result in the integer value that would be calculated by truncating the decimal points of the denary real-number division, in this case `(-14)/4 = -3.5`, and truncating the value `-3.5` gives the value `-3`. In Java, integer division rounds towards zero.

The following equation is always true of integer division, as demonstrated in by the equations the box below it:

`dividend = quotient × divisor + remainder`

```
 i)  14 = 4 ×  3 + 2
ii) -14 = 4 × -3 - 2
```

In Java, the `remainder` will always be the same sign as the `dividend`.

a) Write the results of the equivalent Python commands:

i)	`print(14 // 4)`	
	`print(14 % 4)`	
	`print(14 >> 2)`	
ii)	`print((-14)//4)`	
	`print(14 % 4)`	
	`print((-14)>>2)`	

Python integer division will result in the integer value that would be calculated by the “floor” of the decimal points of the denary real-number division. This means that the value is the next closest integer towards negative infinity ($-\infty$), in this case `(-14)/4 = -3.5`, and the floor, `⌊-3.5⌋`, gives the value `-4`.

Note that the same formula still applies for Python integer division:

`dividend = quotient × divisor + remainder`

```
 i)  14 = 4 ×  3 + 2
ii) -14 = 4 × -4 + 2
```

In Python, the `remainder` will always be a positive number.