Integer division in Haskell: quotRem and divMod

They behave the same if all arguments are positive:

$3 / 5 = 0 + 3/5$

```
div  0  mod 3
quot 0  rem 3
```

$5 / 3 = 1 + 2/3$

```
div  1  mod 2
quot 1  rem 2
```

Negative numbers show the difference:

- quotrem rounds towards 0
- divmod rounds towards negative infinity

$-3 / 5 = 0 - 3/5$

```
div  -1  mod  2
quot  0  rem -3
```

$-5 / 3 = -1 - 2/3$

```
div  -2  mod  1
quot -1  rem -2
```

$3 / -5 = 0 - 3/5$

```
div  -1  mod -2
quot  0  rem  3
```

$5 / -3 = -1 - 2/3$

```
div  -2  mod -1
quot -1  rem  2
```

Rules

```
(quot x y) * y + (rem x y) == x
```

```
(div  x y) * y + (mod x y) == x
```

Usage

- check if a number is divisible: `rem x y == 0`
- use for modular arithmetic (e.g. hours, weekdays): `mod x y`