<table>
<tr><th>Operator</th><th>How it can be used</th><th>Example</th></tr>
<tr><td>% (mod)</td><td>Groups numbers that skip

% n will map numbers that skip by n to the same value</td><td><table>
<tr><td>i</td><td>i%5</td></tr>
<tr><td>0</td><td>0</td></tr>
<tr><td>1</td><td>1</td></tr>
<tr><td>2</td><td>2</td></tr>
<tr><td>3</td><td>3</td></tr>
<tr><td>4</td><td>4</td></tr>
<tr><td>5</td><td>0</td></tr>
<tr><td>6</td><td>1</td></tr>
<tr><td>7</td><td>2</td></tr>
<tr><td>8</td><td>3</td></tr>
<tr><td>9</td><td>4</td></tr>
<tr><td>10</td><td>0</td></tr>
<tr><td>…</td><td>…</td></tr>
</table></td></tr>
<tr><td>/ (int division)</td><td>Groups consecutive numbers

/ n will map n consecutive numbers together to the same value</td><td><table>
<tr><td>i</td><td>i/2</td></tr>
<tr><td>0</td><td>0</td></tr>
<tr><td>1</td><td>0</td></tr>
<tr><td>2</td><td>1</td></tr>
<tr><td>3</td><td>1</td></tr>
<tr><td>4</td><td>2</td></tr>
<tr><td>5</td><td>2</td></tr>
<tr><td>6</td><td>3</td></tr>
<tr><td>7</td><td>3</td></tr>
<tr><td>8</td><td>4</td></tr>
<tr><td>9</td><td>4</td></tr>
<tr><td>10</td><td>5</td></tr>
<tr><td>…</td><td>…</td></tr>
</table></td></tr>
<tr><td>+, -</td><td>Offsets to help the other two tricks work properly

+ n maps values up by n
- n maps values down by n</td><td><table>
<tr><td>i</td><td>(i+1)/2</td></tr>
<tr><td>0</td><td>0</td></tr>
<tr><td>1</td><td>1</td></tr>
<tr><td>2</td><td>1</td></tr>
<tr><td>3</td><td>2</td></tr>
<tr><td>4</td><td>2</td></tr>
<tr><td>5</td><td>3</td></tr>
<tr><td>6</td><td>3</td></tr>
<tr><td>7</td><td>4</td></tr>
<tr><td>8</td><td>4</td></tr>
<tr><td>9</td><td>5</td></tr>
<tr><td>10</td><td>5</td></tr>
<tr><td>…</td><td>…</td></tr>
</table></td></tr>
</table>

Example:

Say we want to map

```
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16…
to
0 1 1 1 2 2 2 3 0 1 1  1  2  2  2  3  0…
```

We start off with the original index i:

```
i              : 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16…
```

The first thing we notice is that the pattern repeats every 8 numbers, which means we want to skip by 8, so the first thing we can try is i % 8:

```
i % 8          : 0 1 2 3 4 5 6 7 0 1 2  3  4  5  6  7  0…
```

Now we notice that there seem to be consecutive numbers grouped in 3’s so the next thing we can try is (i % 8) / 3:

```
(i % 8) / 3    : 0 0 0 1 1 1 2 2 0 0 0  1  1  1  2  2  0…
```

This is close but it looks like we might want to shift here, so let’s take a look at i / 3 and figure out how much we want to shift by:

```
i / 3          : 0 0 0 1 1 1 2 2 2 3 3  3  4  4  4  5  5…
```

This highlighted bit is the pattern that we want, so we should shift by + 2 to get rid of the first two values:

```
i % 8 + 2      : 2 3 4 5 6 7 8 9 2 3 4  5  6  7  8  9  2…
```

```
(i % 8 + 2) / 3: 0 1 1 1 2 2 2 3 0 1 1  1  2  2  2  3  0…
```