Abelian sandpile model

[](http://www.rosettacode.org/wiki/Category:Solutions_by_Programming_Task)

**Abelian sandpile model**  
You are encouraged to [solve this task](http://www.rosettacode.org/wiki/Rosetta_Code:Solve_a_Task) according to the task description, using any language you may know.

|  |
| --- |
| This page uses content from [**Wikipedia**](https://en.wikipedia.org/). The original article was at [Abelian sandpile model](https://en.wikipedia.org/wiki/Abelian_sandpile_model). The list of authors can be seen in the **[page history](https://en.wikipedia.org/w/index.php?title=Abelian+sandpile+model&action=history)**. [As with Rosetta Code](http://www.rosettacode.org/wiki/Rosetta_Code:Copyrights), the text of Wikipedia [is available](https://en.wikipedia.org/wiki/Project:Copyrights" \o "wp:Project:Copyrights) under the [GNU FDL](https://www.gnu.org/licenses/fdl.html). (See links for details on variance) |

Implement the **Abelian sandpile model** also known as **Bak–Tang–Wiesenfeld model**. Its history, mathematical definition and properties can be found under its [wikipedia article](https://en.wikipedia.org/wiki/Abelian_sandpile_model).

The task requires the creation of a 2D grid of arbitrary size on which "piles of sand" can be placed. Any "pile" that has 4 or more sand particles on it *collapses*, resulting in **four particles being subtracted from the pile** and **distributed among its neighbors.**

It is recommended to display the output in some kind of image format, as terminal emulators are usually too small to display images larger than a few dozen characters tall. As an example of how to accomplish this, see the [Bitmap/Write a PPM file](https://rosettacode.org/wiki/Bitmap/Write_a_PPM_file) task.  
[Examples up to 2^30, wow!](http://www.math.cmu.edu/~wes/sand.html)  
[javascript running on web](http://www.natureincode.com/code/various/sandpile.html)  
**Examples:**

0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 1 0 0

0 0 4 0 0 -> 0 1 0 1 0

0 0 0 0 0 0 0 1 0 0

0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 1 0 0

0 0 6 0 0 -> 0 1 2 1 0

0 0 0 0 0 0 0 1 0 0

0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 1 0 0

0 0 0 0 0 0 2 1 2 0

0 0 16 0 0 -> 1 1 0 1 1

0 0 0 0 0 0 2 1 2 0

0 0 0 0 0 0 0 1 0 0