**MakeBricks Solution Code**

MakeBricks is a favorite problem I created for CodingBat, probably because it has more depth than it appears at first. Here I'll walk through one way to solve it.

See also: [MakeBricks Introduction](http://codingbat.com/doc/practice/makebricks-introduction.html) , [Mod Introduction](http://codingbat.com/doc/practice/mod-introduction.html)

Here's the live MakeBricks and the slightly more difficult MakeChocolate problems:   
http://codingbat.com/c1.jpg[makeBricks](http://codingbat.com/prob/p183562?parent=/doc/practice/makebricks-solution-code.html)  http://codingbat.com/c1.jpg[makeChocolate](http://codingbat.com/prob/p191363?parent=/doc/practice/makebricks-solution-code.html)

(in python: http://codingbat.com/c1.jpg[make\_bricks](http://codingbat.com/prob/p118406?parent=/doc/practice/makebricks-solution-code.html)  http://codingbat.com/c1.jpg[make\_chocolate](http://codingbat.com/prob/p190859?parent=/doc/practice/makebricks-solution-code.html) )

* Lots of ways to solve it -- anything without iteration is ok
* For some problems, there is a very short solution -- not always!
* Use "mod" % 5 to figure needed small bricks (vs. loops)
* Think about cases where the goal is **not** possible .. easier

## Fail #1: not enough bricks

* Is the goal bigger than all the big and small bricks we have?
* Easy to express: goal vs. big\*5 + small

## Fail #2: not enough small bricks

* Do we not have enough small bricks to get the last bit of length
* Use goal % 5

## Divide and Conquer

Sometimes a problem appears to be a baffling tangle. Often, if you slow down, you can break it down to 2 or 3 cases, and then solve them one at a time. "Divide and Conquer" as Caesar said, and that's what's going on here, focussing on one aspect of the problem at a time.

[CodingBat.com](http://codingbat.com/) code practice.