



- ✓ The `time()` function (and therefore the preceding randomizing seeding statement) requires the following statement at the beginning of your source code:

```
#include <time.h>
```

- ✓ And, this one:

```
#include <stdlib.h>
```

The Diabolical Dr. Modulus

You may have noticed that the random numbers produced by the `RANDOM` series of programs have been wildly random and often unsuitable. For example:

```
Your turn, BOB.  
You rolled a 23415 on the dice. This lands you on Boardwalk,  
but you've passed Go 585 times, which nets you a  
gross of $117,000!
```

That gets old after a while.

You need to have a way to round down the numbers — a hacksaw to chop off excess values and just give you numbers in a certain range, for example.



Lo, you have a way. It's the mathematical concept *modulus*.

Ah-ooga! Ah-ooga! Mathematical concept alert! Ah-ooga!

Modulus — bah! This term is better known to you as “the remainder,” as in “When you divide 6 into 10, you get a remainder of 4.” Well, 6 gazinta 10 is *modulo 4*. It's written in C as

```
4 = 10 % 6;
```

I don't expect you to remember what a modulus is or does, nor do I expect you to know when to use modulus, or the ablative, *modulo*. The format is the same for any of C's mathematical operators. What's important to know is that you can use the modulus doohickey to pare down larger numbers into smaller, more convenient chunks.

```
modulus = big % little;
```

Read it this way: If you take the huge number `big` and divide it by the smaller number `little`, you get a remainder, which is the `modulo` value.