To call a function on each integer in a (small) range, you **can** use:

```
@results = map { some_func($_) } (5 .. 25);
```

but you should be aware that the `..` operator creates an array of all integers in the range. This can take a lot of memory for large ranges. Instead use:

```
@results = ();
for ($i=5; $i < 500_005; $i++) {
    push(@results, some_func($i));
}
```

This situation has been fixed in Perl5.005. Use of `..` in a `for` loop will iterate over the range, without creating the entire range.

```
for my $i (5 .. 500_005) {
    push(@results, some_func($i));
}
```

will not create a list of 500,000 integers.

20.2.9 How can I output Roman numerals?

Get the http://www.cpan.org/modules/by-module/Roman module.

20.2.10 Why aren't my random numbers random?

If you're using a version of Perl before 5.004, you must call `srand` once at the start of your program to seed the random number generator.

```
BEGIN { srand() if $] < 5.004 }
```

5.004 and later automatically call `srand` at the beginning. Don't call `srand` more than once—you make your numbers less random, rather than more.

Computers are good at being predictable and bad at being random (despite appearances caused by bugs in your programs :-). see the *random* article in the "Far More Than You Ever Wanted To Know" collection in http://www.cpan.org/misc/olddoc/FMTEYEWTK.tgz , courtesy of Tom Phoenix, talks more about this. John von Neumann said, "Anyone who attempts to generate random numbers by deterministic means is, of course, living in a state of sin."

If you want numbers that are more random than `rand` with `srand` provides, you should also check out the Math::TrulyRandom module from CPAN. It uses the imperfections in your system's timer to generate random numbers, but this takes quite a while. If you want a better pseudorandom generator than comes with your operating system, look at "Numerical Recipes in C" at http://www.nr.com/ .

20.2.11 How do I get a random number between X and Y?

`rand($x)` returns a number such that `0 <= rand($x) < $x`. Thus what you want to have perl figure out is a random number in the range from 0 to the difference between your *X* and *Y*.

That is, to get a number between 10 and 15, inclusive, you want a random number between 0 and 5 that you can then add to 10.

```
my $number = 10 + int rand( 15-10+1 );
```