

Re: rand and srand

Source: http://coding.derkeiler.com/Archive/C_CPP/comp.lang.c/2008-03/msg01304.html

- *From:* "Bill Cunningham" <nospam@xxxxxxxxxx>
 - *Date:* Sun, 09 Mar 2008 21:14:44 GMT
-

The function `rand()` returns a value between 0 and `RAND_MAX`, not between 0 and what you pass to `srand()`. The latter function only sets the seed for the random number generator (on which the next random number will be based).

If you want to have a non-predictable sequence of random numbers, it's often a good idea to set the seed to something that is not a literal and will change each time the program is executed.

Additionally, if you want to generate a random number between 0 and 2000, the code to do that uses the modulus operator, thusly:

```
printf("%u\n", (rand() % 2001));
```

For one, without the `\n` (newline), nothing will separate your numbers from each other. The second part of this is the `% 2001`. That divides the value from `rand()` by 2001, and returns the remainder – so the number can be anything from 0 to 2000. If `rand()` returns 2001, the modulus makes it 0, etc.

Thanks so much for your advice. I haven't used code with modulus yet and this is the first time I've seen it.

As for your `main()`, I wasn't actually aware that such code was legal C. My guess is that it is actually `void main()`, because you never return a value from it, so the exit status of your program is most likely undefined. It is a good idea, particularly in UNIX, to declare it `int`, and return 0 at the end (unless the program failed).

The C99 standard wants `int`. But all the compilers I've used accept `main()` and `main(int argc, char *argv[])` when parameters are used. Otherwise automatic void of course with just `()` as parameters. Thanks for the advice on `rand`. So is there really a need to use `srand ()` ?

Re: rand and srand

Bill

.