Example 82. (bonus!) The LFSR $x_{n+31} \equiv x_{n+28} + x_n \pmod{2}$ from Example 56, which is used in glibc, is entirely predictable because observing $x_1, x_2, \ldots, x_{31}$ we know what $x_{32}, x_{33}, \ldots$ are going to be. Alice tries to reduce this predictability by using only $x_3, x_6, x_9, \ldots$ as the output of the LFSR. Demonstrate that this PRG is still perfectly predictable by showing the following:

Challenge. Find a simple LFSR which produces $x_3, x_6, x_9, \ldots$

Send me the LFSR, and an explanation how you found it, by next week for a bonus point!

Comment. There is nothing special about this LFSR. Moreover, a generalization of this argument shows that only outputting every Nth bit of an LFSR is always going to result in an entirely predictable PRG.

The Blum-Blum-Shub PRG

The Blum-Blum-Shub PRG is an example of a PRG, which is believed to be unpredictable.

More precisely, it has been shown that the ability to predict its values is equivalent to being able to efficiently solve the quadratic residuosity problem (which is believed to be hard). Currently, the best way to "solve" the quadratic residuosity problem mod M relies on factoring M. However, factoring large numbers is considered to be hard (and lots of crypto relies on that).

Quadratic residuosity problem. Given big $M = pq$ and a residue x modulo M, decide whether x is a quadratic residue. (About $M/4$ are quadratic residues (the exact number is $\phi(M)/4 = (p-1)(q-1)/4$); $M/2$ are easily determined to be nonsquare using the Jacobi symbol [don't worry if you haven't heard about that].)

(Blum-Blum-Shub PRG) Let $M = pq$ where p, q are large primes $\equiv 3 \pmod{4}$.

From the seed y_0, we generate $y_{n+1} \equiv y_n^2 \pmod{M}$.

The random bits x_n we produce are $y_n \pmod{2}$ (i.e. $x_n = \text{least bit of}(y_n)$).

B-B-S is very slow, and mostly of theoretical value. However, as indicated above, it is interesting because it is indeed unpredictable (to anyone not knowing the factorization of M) if an important number theory problem (the quadratic residuosity problem) is "hard" (this can be made precise), as is believed to be the case.

Why the conditions on p and q? Recall from the CRT that an invertible quadratic residue x^2 modulo $M = pq$ has exactly four squareroots $\pm x$, $\pm y$. The condition $3 \pmod{4}$ guarantees that, of these four, exactly one is itself a quadratic residue. As a consequence, the mapping $y \mapsto y^2 \pmod{M}$ is 1-1 when restricted to invertible quadratic residues (see below).

Comment. For obvious reasons, the seed $y_0 \equiv \pm 1 \pmod{M}$ should be excluded. Also, for the above considerations to apply, the seed needs to be coprime to M. However, we don't need to worry about that: running into a factor of M by accident is close to impossible (recall that nobody should be able to factor M even on purpose and with lots of time and resources).

Comment. To increase speed, at the expense of some security, we can also take several, say k, bits of y_n (as long as k is small, say, $k \leqslant \log_2 \log_2 M$).

Example 83. Generate random bits using the B-B-S PRG with $M = 77$ and seed 3.

Solution. With $y_0 = 3$, we have $y_1 \equiv y_0^2 = 9$, followed by $y_2 \equiv y_1^2 \equiv 4 \pmod{77}$, $y_3 \equiv 16$, $y_4 \equiv 25$, $y_5 \equiv 9$, so that the values y_n now start repeating.

These numbers are, however, not the output of the PRG. We only output the least bit of the numbers y_n, i.e. the value of $y_n \pmod{2}$. For $y_1 \equiv 9$ we output 1, for $y_2 \equiv 4$ we output 0, for $y_3 \equiv 16$ we output 0, for $y_4 \equiv 25$ we output 1, and so on.

In other words, the seed 3 produces the sequence $1, 0, 0, 1, 1, 0, 0, 1, 1, 0, \ldots$ of period 4.

Comment. Note that it was completely to be expected that the numbers repeat. In fact, we immediately see that the number of possible y_n is at most the number of invertible quadratic residues, of which there are only $\phi(77)/4 = 15$.

Example 84.

(a) List all invertible quadratic residues modulo 21. Compute the square of all these residues.

(b) Repeat the first part modulo 33 and modulo 35. When computing the squares of these, do you notice a difference modulo 35?

[Note that $35 = 5 \cdot 7$ with $5 \equiv 1 \pmod{4}$. This case is excluded in the B-B-S PRG.]

Solution. (final answers only)

(a) Among the $\phi(21) = 12$ many invertible elements, the squares are $1, 4, 16$ (exactly a quarter).
Computing their squares: $1^2 \equiv 1$, $4^2 \equiv 16$, $16^2 \equiv 4 \pmod{21}$. Note that the squares are all different!

(b) Modulo 33: among the $\phi(33) = 20$ many invertible elements, the squares are $1, 4, 16, 25, 31 \equiv 8^2$ (exactly a quarter). Computing their squares: $1^2 \equiv 1$, $4^2 \equiv 16$, $16^2 \equiv 25$, $25^2 \equiv 31$, $31^2 \equiv 4 \pmod{33}$. Again, all the squares are different!

Modulo 35: among the $\phi(35) = 24$ many invertible elements, the squares are 1, 4, 9, $11 \equiv 9^2$, 16, $29 \equiv 8^2$ (exactly a quarter). Computing their squares: $1^2 \equiv 1$, $4^2 \equiv 16$, $9^2 \equiv 11$, $11^2 \equiv 16$, $16^2 \equiv 11$, $29^2 \equiv 1 \pmod{35}$. Observe that these are not all different: for instance, $9^2 \equiv 16^2 \pmod{35}$.

Advanced comment. The map $x \mapsto x^2 \pmod{p}$ restricted to invertible quadratic residues is 1-1 if and only if -1 is not a quadratic residue (which, by the next result, is equivalent to $p \equiv 3 \pmod{4}$).

[*Sketch of proof.* The map is 1-1 if and only if, for each invertible quadratic residue x^2, exactly one of the two square roots $\pm x$ is itself a quadratic residue. This is equivalent to -1 not being a quadratic residue.

Indeed, if -1 is a quadratic residue, then x and $-x$ are either both quadratic residues or both not.

On the other hand, if not exactly one of $\pm x$ is a quadratic residue then, because exactly half of the invertible residues are quadratic, there would be some pair of residues $\pm z$ which are both quadratic. But then $-z$ and z^{-1} are quadratic residues which implies that their product $-zz^{-1} \equiv -1$ would be a quadratic residue as well.]

Theorem 85. -1 is a quadratic residue modulo (an odd prime) p if and only if $p \equiv 1 \pmod{4}$.

In other words, the quadratic congruence $x^2 \equiv -1 \pmod{p}$ has a solution if and only if $p \equiv 1 \pmod{4}$.

Solution. Let us first see that $p \equiv 1 \pmod{4}$ is necessary. Assume $x^2 \equiv -1 \pmod{p}$. Then, by Fermat's little theorem, $x^{p-1} \equiv 1 \pmod{p}$. On the other hand, $x^{p-1} = (x^2)^{(p-1)/2} \equiv (-1)^{(p-1)/2} \pmod{p}$. We therefore need $(-1)^{(p-1)/2} = 1$, which is equivalent to $(p-1)/2$ being even. Which is equivalent to $p \equiv 1 \pmod{4}$. (Make sure that's absolutely clear!)

On the other hand, assume that $p \equiv 1 \pmod{4}$. We will show that $x = \left(\frac{p-1}{2}\right)!$ has the property that $x^2 \equiv -1 \pmod{p}$. Indeed,

$$\left[\left(\frac{p-1}{2}\right)!\right]^2 = (-1)^{(p-1)/2}\left(1 \cdot 2 \cdot \ldots \cdot \frac{p-1}{2}\right)^2 = (\pm 1) \cdot (\pm 2) \cdot \ldots \cdot \left(\pm \frac{p-1}{2}\right) \equiv -1 \pmod{p}.$$

[Here, $(\pm 1) \cdot (\pm 2) \cdots$ is short for $1 \cdot (-1) \cdot 2 \cdot (-2) \cdots$.] For the final congruence, observe that $\pm 1, \pm 2, \ldots, \pm \frac{p-1}{2}$ is a complete set of all nonzero residues. When multiplying all residues, each will cancel with its (modular) inverse, except the ones that are their own inverse. But $a \cdot a \equiv 1 \pmod{p}$ has only the solution $a \equiv \pm 1$, so that ± 1 are the only residues not canceling.

Comment. The final step of our argument is known as Wilson's congruence: $(p-1)! \equiv -1 \pmod{p}$.

Theorem 86. (advanced) Let $M = pq$ where p, q are primes $\equiv 3 \pmod{4}$. Then the sequence generated by $y_{n+1} \equiv y_n^2 \pmod{M}$ repeats with period dividing $\operatorname{lcm}(\phi(p-1), \phi(q-1))$.

In particular, the period of the corresponding B-B-S PRG divides $\operatorname{lcm}(\phi(p-1), \phi(q-1))$.

Proof.

- Observe that the numbers are $y_n = y_{n-1}^2 = y_{n-2}^4 = \ldots = y_0^{2^n} \pmod{M}$. Hence, $y_n \equiv y_0^{2^n} \pmod{M}$.
- Instead of determining the period directly modulo $M = pq$, we determine the periods modulo p and q. [Why? By the CRT, $y_m \equiv y_n \pmod{M}$ if and only if $y_m \equiv y_n \pmod{p}$ and $y_m \equiv y_n \pmod{q}$.]
  The period modulo M then is the lcm of of the two periods modulo p and q.
- $y_m \equiv y_n \pmod{p}$
  $\iff \quad y_0^{2^m} \equiv y_0^{2^n} \pmod{p}$
  $\Longleftarrow \quad 2^m \equiv 2^n \pmod{\phi(p)}$
  [it would be "$\iff$" with $2^m \equiv 2^n \pmod{k}$ where k is the order of $y_0 \pmod{p}$]
  $\iff \quad 2^m \equiv 2^n \pmod{p-1}$
  [note that 2 is not invertible $\pmod{p-1}$; but 2 is invertible $\left(\operatorname{mod} \frac{p-1}{2}\right)$ because $p \equiv 3 \pmod{4}$]
  $\iff \quad 2^{m-1} \equiv 2^{n-1} \left(\operatorname{mod} \frac{p-1}{2}\right)$ [note that $m, n \geqslant 1$]
  $\Longleftarrow \quad m \equiv n \left(\operatorname{mod} \phi\left(\frac{p-1}{2}\right)\right)$
  [again, it would be "$\iff$" with $m \equiv n \pmod{k}$ where k is the order of $2 \left(\operatorname{mod} \frac{p-1}{2}\right)$]
- Reading this backwards, we see that the sequence y_n modulo p repeats after $\phi\left(\frac{p-1}{2}\right)$ terms.
  In other words, the (minimal) period divides $\phi\left(\frac{p-1}{2}\right) = \phi(p-1)$.

  Comment. Here we used $p \equiv 3 \pmod{4}$ to conclude $\phi\left(\frac{p-1}{2}\right) = \phi(p-1)$. Indeed, in that case, $p-1$ is divisible by 2 but not by 4. Hence, 2 and $\frac{p-1}{2}$ are coprime, so that $\phi(p-1) = \phi(2)\phi\left(\frac{p-1}{2}\right) = \phi\left(\frac{p-1}{2}\right)$.
- By the CRT, the period modulo $M = pq$ divides $\operatorname{lcm}(\phi(p-1), \phi(q-1))$. □

Example. In Example 83, we had $M = 7 \cdot 11$, so that the period of the PRG must divide $\operatorname{lcm}(\phi(6), \phi(10)) = \operatorname{lcm}(2, 4) = 4$.

Comment. In practice, people therefore say that, for the cycle length of B-B-S to be large, $\gcd(\phi(p-1), \phi(q-1))$ should be small.

Example 87. We mentioned that the unpredictability of the B-B-S PRG relies on the difficulty of factoring large numbers. Here's an indication how difficult it seems to be. In 1991, RSA Laboratories challenged everyone to factor several numbers including:

```
13506641086599522334960321627880596993888147560566702752448514385152651060485953383394028715057190944179820728216447155137368041970396419174\
304649658927425623934102086438320211037295872576235850964311056407350150818751067659462920556368552947521350085287941637732853390610975054433499981115005697723689092756
```

Since then, nobody has been able to factor this 1024 bit number (309 decimal digits). Until 2007, cash prizes were offered up to 200,000 USD, with 100,000 USD for the number above.

https://en.wikipedia.org/wiki/RSA_Factoring_Challenge

Let us illustrate how to actually use this number in the B-B-S PRG.

```
Sage] rsa = Integer("1350664108659952233496032162788059699388814756056670275244851438515\
                     2651060485953383394028715057190944179820728216447155137368041970396419174\
                     3046496589274256239341020864383202110372958725762358509643110564073501\
                     5081875106765946292055636855294752135008528794163773285339061097505443\
                     3499981115005697723689092756")
```

Sage] seed = randint(2,rsa-2)

Sage] y = seed; prg = []

Sage] for i in [1..25]:
          y = power_mod(y, 2, rsa)
          prg.append(y % 2)

Sage] prg

$$[0,1,0,1,1,1,0,0,0,0,0,0,0,0,0,1,0,1,0,0,1,0,1,0,1]$$

If you are able, even after gigabytes of pseudorandom bits, to predict the next bits with an accuracy better than 50% (which is just pure guessing), then you likely have a shot at factoring the big integer. You would be the first!

Of course, it is not impressive to see a few random bits in the example above. After all, the seed (which you don't know!) itself consists of 1024 random bits. The whole point is that we can, from these 1024 random bits, produce gigabytes of further pseudorandom bits. As of this day, no one would be able to distinguish these from truly random bits.

While all of this works nicely, B-B-S is considered to be too slow for most practical purposes.

Comment. Note that $M = 135...563 \equiv 3 \pmod{4}$. Hence it cannot be a product of primes p, q which are both $3 \pmod{4}$ (because $3 \cdot 3 \equiv 1 \pmod{4}$).

Example 88. (extra) Generate random bits using the B-B-S PRG with $M = 209$ and seed 10. What is the period of the generated sequence? (Then repeat with seed 25.)

Solution. (final answer only) The seed 10 produces the sequence $0, 1, 0, 1, 1, 1, \ldots$ of period 6.

The seed 25 generates the sequence $1, 0, 0, 1, 1, 0, 1, 0, 0, 0, 1, 1, \ldots$ of period 12.

[By the way, it is a good idea to let Sage assist you.]