In the context of pseudorandom number generation, the values for the constants **a** and **b** in a linear congruential generator (LCG) can dramatically affect the quality of the randomness produced. The general form of the pseudorandom generator you're referring to is:

cur=(a⋅cur+b)mod  n\text{cur} = (a \cdot \text{cur} + b) \mod ncur=(a⋅cur+b)modn

where:

* cur is the current pseudorandom value,
* a and b are constants (the multiplier and increment, respectively),
* n is the modulus (which in this case is n = 1000),
* The initial value of cur (the seed) is typically given.

For the generator to **not look very random**, we can choose values for **a** and **b** that result in poor statistical properties like repetitions, short cycles, or predictable patterns.

**Examples of Poor a and b Values:**

**1. a = 1 and b = 0**

In this case, the next value of cur will always be equal to the current value, as:

cur=(1⋅cur+0)mod  1000=curmod  1000\text{cur} = (1 \cdot \text{cur} + 0) \mod 1000 = \text{cur} \mod 1000cur=(1⋅cur+0)mod1000=curmod1000

This means that the sequence of values generated will always be the same, making it **completely deterministic** and **not random**.

* **Effect:** The sequence will be constant and predictable. For example, if the initial value is 42, the entire sequence will always be 42.

**2. a = 2 and b = 0**

Here, the multiplication by 2 causes a pattern in the numbers, particularly if the current value (cur) starts as a power of 2. Since 2 is not a full-period multiplier for modulo 1000, the sequence will repeat after a certain number of steps, and the numbers will follow a predictable doubling pattern.

cur=(2⋅cur+0)mod  1000\text{cur} = (2 \cdot \text{cur} + 0) \mod 1000cur=(2⋅cur+0)mod1000

* **Effect:** The sequence will grow by powers of 2 (modulo 1000). If the seed is a power of 2 (like 256), the sequence will follow a repeating pattern of powers of 2 modulo 1000.

**3. a = 500 and b = 0**

In this case, multiplying by 500 results in a sequence that will bounce back and forth between a small set of values.

cur=(500⋅cur+0)mod  1000\text{cur} = (500 \cdot \text{cur} + 0) \mod 1000cur=(500⋅cur+0)mod1000

* **Effect:** The sequence might exhibit a repetitive or periodic behavior that isn’t very random, especially if the seed is not a large number. Values will jump in regular intervals based on the modulus 1000.

**4. a = 999 and b = 999**

In this case, the multiplier and increment are both close to n, which can cause the sequence to have a small cycle or to converge to a small set of values quickly.

cur=(999⋅cur+999)mod  1000\text{cur} = (999 \cdot \text{cur} + 999) \mod 1000cur=(999⋅cur+999)mod1000

* **Effect:** This choice can result in values that become very repetitive or cycle too quickly, especially with certain starting seeds. It doesn’t take full advantage of the entire modulus range, making the sequence less random.

**Why These Choices are Not Random:**

* **Small Cycle Length:** If a and b are too simple (like a = 1 or a = 2), the sequence will quickly fall into a short cycle or even be constant.
* **Predictability:** Multiplying by small values or using increments of 0 means the sequence is highly predictable, leading to poor randomness.
* **Not Full Periods:** Some values for a and b result in the sequence repeating before reaching a full period, reducing the number of possible unique values the generator can produce.

**Summary:**

The following combinations of a and b will result in sequences that do not look very random:

* **a = 1, b = 0** → Always the same number.
* **a = 2, b = 0** → Doubling pattern, not random.
* **a = 500, b = 0** → Sequence jumps in small intervals, may have a short period.
* **a = 999, b = 999** → Quickly repeats or converges to a small set of values.

To ensure better randomness, a and b should be chosen carefully to avoid these patterns, and a larger n with well-chosen parameters can help create a longer, more random sequence.