--- Day 23: Coprocessor Conflagration ---

You decide to head directly to the CPU and fix the printer from there. As you get close, you find an experimental coprocessor doing so much work that the local programs are afraid it will halt and catch fire. This would cause serious issues for the rest of the computer, so you head in and see what you can do.

The code it's running seems to be a variant of the kind you saw recently on that tablet. The general functionality seems very similar, but some of the instructions are different:

- `set X Y` sets register `X` to the value of `Y`.
- `sub X Y` decreases register `X` by the value of `Y`.
- `mul X Y` sets register `X` to the result of multiplying the value contained in register `X` by the value of `Y`.
- `jnz X Y` jumps with an offset of the value of `Y`, but only if the value of `X` is not zero. (An offset of `2` skips the next instruction, an offset of `-1` jumps to the previous instruction, and so on.)

Only the instructions listed above are used. The eight registers here, named `a` through `h`, all start at `0`.

The coprocessor is currently set to some kind of debug mode, which allows for testing, but prevents it from doing any meaningful work.

If you run the program (your puzzle input), how many times is the `mul` instruction invoked?

To begin, get your puzzle input.

Answer: [Submit]

You can also [Share] this puzzle.