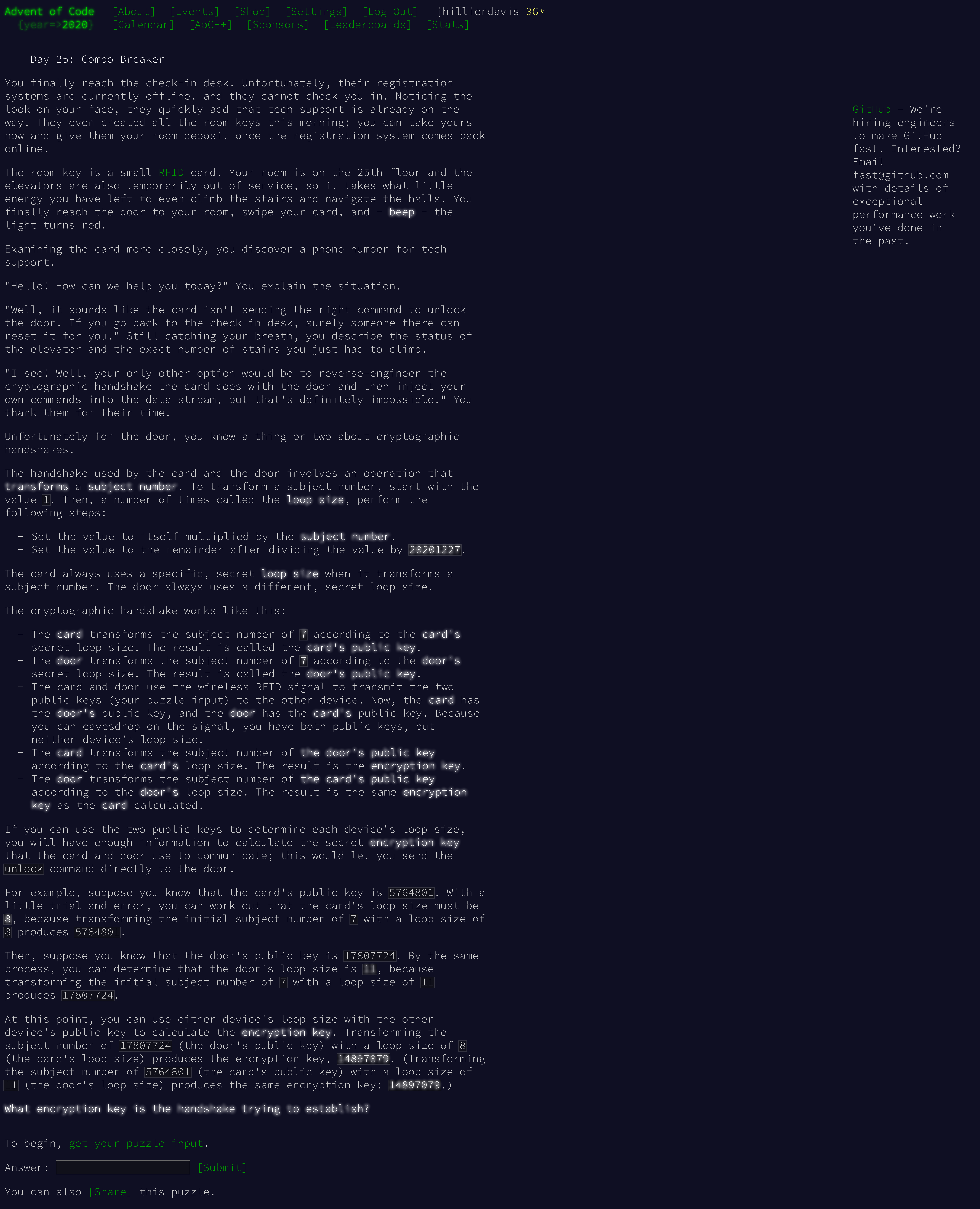

--- Day 25: Combo Breaker ---

You finally reach the check-in desk. Unfortunately, their registration systems are currently offline, and they cannot check you in. Noticing the look on your face, they quickly add that tech support is already on the way! They even created all the room keys this morning; you can take yours now and give them your room deposit once the registration system comes back online.

The room key is a small RFID card. Your room is on the 25th floor and the elevators are also temporarily out of service, so it takes what little energy you have left to even climb the stairs and navigate the halls. You finally reach the door to your room, swipe your card, and - **beep** - the light turns red.

Examining the card more closely, you discover a phone number for tech support.

"Hello! How can we help you today?" You explain the situation.

"Well, it sounds like the card isn't sending the right command to unlock the door. If you go back to the check-in desk, surely someone there can reset it for you." Still catching your breath, you describe the status of the elevator and the exact number of stairs you just had to climb.

"I see! Well, your only other option would be to reverse-engineer the cryptographic handshake the card does with the door and then inject your own commands into the data stream, but that's definitely impossible." You thank them for their time.

Unfortunately for the door, you know a thing or two about cryptographic handshakes.

The handshake used by the card and the door involves an operation that **transforms** a **subject number**. To transform a subject number, start with the value `1`. Then, a number of times called the **loop size**, perform the following steps:

- Set the value to itself multiplied by the **subject number**.
- Set the value to the remainder after dividing the value by `20201227`.

The card always uses a specific, secret **loop size** when it transforms a subject number. The door always uses a different, secret loop size.

The cryptographic handshake works like this:

- The **card** transforms the subject number of `7` according to the **card's** secret loop size. The result is called the **card's public key**.
- The **door** transforms the subject number of `7` according to the **door's** secret loop size. The result is called the **door's public key**.
- The card and door use the wireless RFID signal to transmit the two public keys (your puzzle input) to the other device. Now, the **card** has the **door's** public key, and the **door** has the **card's** public key. Because you can eavesdrop on the signal, you have both public keys, but neither device's loop size.
- The **card** transforms the subject number of **the door's public key** according to the **card's** loop size. The result is the **encryption key**.
- The **door** transforms the subject number of **the card's public key** according to the **door's** loop size. The result is the same **encryption key** as the **card** calculated.

If you can use the two public keys to determine each device's loop size, you will have enough information to calculate the secret **encryption key** that the card and door use to communicate; this would let you send the `unlock` command directly to the door!

For example, suppose you know that the card's public key is `5764801`. With a little trial and error, you can work out that the card's loop size must be **`8`**, because transforming the initial subject number of `7` with a loop size of `8` produces `5764801`.

Then, suppose you know that the door's public key is `17807724`. By the same process, you can determine that the door's loop size is **`11`**, because transforming the initial subject number of `7` with a loop size of `11` produces `17807724`.

At this point, you can use either device's loop size with the other device's public key to calculate the **encryption key**. Transforming the subject number of `17807724` (the door's public key) with a loop size of `8` (the card's loop size) produces the encryption key, **`14897079`**. (Transforming the subject number of `5764801` (the card's public key) with a loop size of `11` (the door's loop size) produces the same encryption key: **`14897079`**.)

What encryption key is the handshake trying to establish?

To begin, get your puzzle input.

Answer: [Submit]

You can also [Share] this puzzle.