Advent of Code   [About]   [Events]   [Shop]   [Settings]   [Log Out]   Arthur 6*
0xffff&2022   [Calendar]   [AoC++]   [Sponsors]   [Leaderboard]   [Stats]

--- Day 6: Tuning Trouble ---

The preparations are finally complete; you and the Elves leave camp on foot and begin to make your way toward the star fruit grove.

As you move through the dense undergrowth, one of the Elves gives you a handheld device. He says that it has many fancy features, but the most important one to set up right now is the communication system.

However, because he's heard you have significant experience dealing with signal-based systems, he convinced the other Elves that it would be okay to give you their one malfunctioning device - surely you'll have no problem fixing it.

As if inspired by comedic timing, the device emits a few colorful sparks.

To be able to communicate with the Elves, the device needs to lock on to their signal. The signal is a series of seemingly-random characters that the device receives one at a time.

To fix the communication system, you need to add a subroutine to the device that detects a start-of-packet marker in the datastream. In the protocol being used by the Elves, the start of a packet is indicated by a sequence of four characters that are all different.

The device will send your subroutine a datastream buffer (your puzzle input); your subroutine needs to identify the first position where the four most recently received characters were all different. Specifically, it needs to report the number of characters from the beginning of the buffer to the end of the first such four-character marker.

For example, suppose you receive the following datastream buffer:

mjqjpqmgbljsphdztnvjfqwrcgsmlb

After the first three characters (mjq) have been received, there haven't been enough characters received yet to find the marker. The first time a marker could occur is after the fourth character is received, making the most recent four characters mjqj. Because j is repeated, this isn't a marker.

The first time a marker appears is after the seventh character arrives. Once it does, the last four characters received are jpqm, which are all different. In this case, your subroutine should report the value 7, because the first start-of-packet marker is complete after 7 characters have been processed.

Here are a few more examples:

  - bvwbjplbgvbhsrlpgdmjqwftvncz: first marker after character 5
  - nppdvjthqldpwncqszvftbrmjlhg: first marker after character 6
  - nznrnfrfntjfmvfwmzdfjlvtqnbhcprsg: first marker after character 10
  - zcfzfwzzqfrljwzlrfnpqdbhtmscgvjw: first marker after character 11

How many characters need to be processed before the first start-of-packet marker is detected?

Your puzzle answer was 1100.

The first half of this puzzle is complete! It provides one gold star: *

--- Part Two ---

Your device's communication system is correctly detecting packets, but
still isn't working. It looks like it also needs to look for messages.

A start-of-message marker is just like a start-of-packet marker, except
it consists of 14 distinct characters rather than 4.

Here are the first positions of start-of-message markers for all of the
above examples:

  - mjqjpqmgbljsphdztnvjfqwrcgsmlb: first marker after character 19
  - bvwbjplbgvbhsrlpgdmjqwftvncz: first marker after character 23
  - nppdvjthqldpwncqszvftbrmjlhg: first marker after character 23
  - nznrnfrfntjfmvfwmzdfjlvtqnbhcprsg: first marker after character 29
  - zcfzfwzzqfrljwzlrfnpqdbhtmscgvjw: first marker after character 26

How many characters need to be processed before the first start-of-
message marker is detected?

Answer: [   ] [Submit]

Although it hasn't changed, you can still get your puzzle input.

You can also [Share] this puzzle.