

[Advent of Code](#)
[\[About\]](#)
[\[AoC++\]](#)
[\[Events\]](#)
[\[Settings\]](#)
[\[Log Out\]](#)
[Harry Comp 32\\*](#)  
[{'year':2016}](#)
[\[Calendar\]](#)
[\[Leaderboard\]](#)
[\[Stats\]](#)
[\[Sponsors\]](#)

--- Day 21: Scrambled Letters and Hash ---

The computer system you're breaking into uses a weird scrambling function to store its passwords. It shouldn't be much trouble to create your own scrambled password so you can add it to the system; you just have to implement the scrambler.

The scrambling function is a series of operations (the exact list is provided in your puzzle input). Starting with the password to be scrambled, apply each operation in succession to the string. The individual operations behave as follows:

Our [sponsors](#) help make AoC possible:

[A&K Rentals](#) - Affordable, high-quality homes just north of Kansas City.

- `swap position X with position Y` means that the letters at indexes `X` and `Y` (counting from `0`) should be swapped.
- `swap letter X with letter Y` means that the letters `X` and `Y` should be swapped (regardless of where they appear in the string).
- `rotate left/right X steps` means that the whole string should be rotated; for example, one right rotation would turn `abcd` into `dabc`.
- `rotate based on position of letter X` means that the whole string should be rotated to the right based on the index of letter `X` (counting from `0`) as determined before this instruction does any rotations. Once the index is determined, rotate the string to the right one time, plus a number of times equal to that index, plus one additional time if the index was at least `4`.
- `reverse positions X through Y` means that the span of letters at indexes `X` through `Y` (including the letters at `X` and `Y`) should be reversed in order.
- `move position X to position Y` means that the letter which is at index `X` should be removed from the string, then inserted such that it ends up at index `Y`.

For example, suppose you start with `abcde` and perform the following operations:

- `swap position 4 with position 0` swaps the first and last letters, producing the input for the next step, `ebcda`.
- `swap letter d with letter b` swaps the positions of `d` and `b`: `edcba`.
- `reverse positions 0 through 4` causes the entire string to be reversed, producing `abcde`.
- `rotate left 1 step` shifts all letters left one position, causing the first letter to wrap to the end of the string: `bcdea`.
- `move position 1 to position 4` removes the letter at position `1` (`c`), then inserts it at position `4` (the end of the string): `bdeac`.
- `move position 3 to position 0` removes the letter at position `3` (`a`), then inserts it at position `0` (the front of the string): `abdec`.
- `rotate based on position of letter b` finds the index of letter `b` (`1`), then rotates the string right once plus a number of times equal to that index (`2`): `ecabd`.

- `rotate based on position of letter d` finds the index of letter `d` (4), then rotates the string right once, plus a number of times equal to that index, plus an additional time because the index was at least 4, for a total of 6 right rotations: `decab`.

After these steps, the resulting scrambled password is `decab`.

Now, you just need to generate a new scrambled password and you can access the system. Given the list of scrambling operations in your puzzle input, what is the result of scrambling `abcdefgh`?

Your puzzle answer was `agcebfhd`.

--- Part Two ---

You scrambled the password correctly, but you discover that you **can't actually modify** the **password file** on the system. You'll need to un-scramble one of the existing passwords by reversing the scrambling process.

What is the un-scrambled version of the scrambled password `fbgdceah`?

Your puzzle answer was `afhdbegc`.

Both parts of this puzzle are complete! They provide two gold stars: \*\*

At this point, you should **return to your advent calendar** and try another puzzle.

If you still want to see it, you can **get your puzzle input**.

You can also **[Share]** this puzzle.