Hexegic Coding Exercise

Please write a command-line program `rotate` that takes three arguments:

- the word `left` or `right`
- an input filename
- an output filename

to be run as e.g. `rotate left in-file out-file` or `rotate right in-file out-file`.

The program should write an output file that contains the contents of the input file rotated either one bit left, or one bit right, depending on the first argument.

To clarify "rotated one bit", if the input file's contents are a stream of bits $b_0 b_1 \dots b_M b_N$, where b_0 is the most significant bit of the first byte of the input file, and b_N is the least significant bit of the last byte of the input file:

- Rotated one bit left means that the output file is $b_1 \dots b_M b_N b_0$.
- Rotated one bit right means that the output file is $b_N b_0 b_1 \dots b_M$.

Notes:

- Programming language: Rust, Go, C or C++, at your preference.
- Platform: Linux or Windows at your preference.
- Your implementation should embody your understanding of production-quality, security-critical code.
- Please don't spend more than an hour or two on it.
- The above description is intended to be unambiguous and not be any sort of trick question: if you're at all unclear about what we're asking you to do then please ask.
- Although the role requires Rust and Go on Linux, please don't let that bias your selection of language or platform, unless you're confident in your knowledge of those languages and/or platform.