Describe how to simulate an arbitrary Turing machine to make it ***error-tolerant***. Specifically, given an arbitrary Turing machine M, describe a new Turing machine M' that accepts and rejects exactly the same strings as M, even though an evil pixie named Lenny will move the head of M' to an ***arbitrary*** location on the tape some finite number of *unknown* times during the execution of M'.

You do not have to describe M' in complete detail, but do give enough details that a seasoned Turing machine programmer could work out the remaining mechanical details.

As stated, this problem has no solution! If M halts on all inputs after a finite number of steps, then Lenny can make any substring of the input string completely invisible to M. For example, if the true input string is `INPUT-STRING`, Lenny can make M believe the input string is actually `IMPING`, by moving the head to the second `I` whenever it tries to move to `R`, and by moving the head to `P` when it tries to move to `U`. Because M halts after a finite number of steps, Lenny only has a finite number of opportunities to move the head.

In fact, with more care, Lenny can make M think the input string is *any* string that uses only symbols from the actual input string; if the true input string is `INPUT-STRING`, Lenny can make M believe the input string is actually `GRINNING-PUTIN-IS-GRINNING`.)

However, there are several different ways to rescue the problem. For each of the following restrictions on Lenny's behavior, and for any Turing machine M, one can design a Turing machine M' that simulates M despite Lenny's interference.

- Lenny can move the head only a *bounded* number of times. For example: Lenny can move the head at most 374 times.
- Whenever Lenny moves the head, he changes the state of the machine to a special error state `lenny`.
- Whenever Lenny moves the head, he moves it to the left end of the tape.
- Whenever Lenny moves the head, he moves it to a blank cell to the right of all non-blank cells.
- Whenever Lenny moves the head, he moves it to a cell containing a particular symbol in the input alphabet, say `0`.
