We can construct a multi-tape Turing machine to solve this in . Our Turing machine will have an alphabet of {0,1,□, ▷} plus natural numbers 0 through the total number of lines in the program, an input tape (containing whatever is the input to the program), a work tape we shall call , another work tape we’ll call , and an output tape.

First, we pre-process by copying the first cells of the input tape to (the start symbol ▷ plus the -long input). will function as the array A in the program. The reason we had to copy the input tape to is because we need to be able to edit it, which we can’t do to the input tape because input tapes are read-only.

Next, we simulate the program. Only one cell in is in use, and it displays the current label the program is in. The register’s heads point to the th cell of (whatever is initialized to in the program), cell in (the only relevant cell in that tape), and the cell of the input and output tapes.