

## Solutions

---

### PAPIR

The key observation is that no folding can harm us in achieving the shortest possible folded piece of paper: any legal folding can be thought of as cutting off an adequate number of pieces off the left or right side of the paper.

What remains is to find a legal sequence of moves as efficiently as possible. One suitable algorithm always tries to cut off the shortest amount of paper possible, in  $O(N^2)$  time.

### ZMAJ

After observing that we can model the variables as a vector, and assignment operations as matrices, we can use the tools of linear algebra to efficiently solve the problem.

For example, consider a program containing four assignment operations symbolically dubbed A, B, C and D. A and B are contained in a REPEAT 3 loop, after which there is a REPEAT 4 loop containing assignments C and D. Let the variable values before executing the statements be contained in the vector  $x$ .

Then, after executing the two loops, the vector holding variable values will be equal to:

$$A \cdot B \cdot A \cdot B \cdot A \cdot B \cdot C \cdot D \cdot C \cdot D \cdot C \cdot D \cdot C \cdot D \cdot x$$

or

$$(A \cdot B)^3 \cdot (C \cdot D)^4 \cdot x$$

Since the assignment matrices are quadratic, the above expressions yield a vector.

Hence, we can model all loops as matrices raised to some power and used logarithmic matrix exponentiation yielding an  $O(\text{lines\_of\_code} \cdot \text{print\_statements} \cdot \text{variables}^3)$  algorithm with a constant factor of  $\log_2(100000)$ .