ED – exercise 10 – 2024.04.16

Zipping lists

GOAL

Pointer juggling, practice for exam exercise #1.

INSTRUCTIONS

Log into http://ed.fdi.ucm.es/domjudge/team with your credentials. Make sure to **submit at least 1 answer before the period ends**, even if it does not work. Before leaving the classroom, even if your program seems to work, **go by the teacher's desk** to see if it can be improved.

Problem statement

Sometimes you need to combine two (or more) lists into a single list, in round-robin fashion (taking the 1st element of the 1st list, 2nd of the 2nd list, and so on). This operation is called 'zipping' lists, and is quite common in, say, Python. Implement it using pointer-juggling.

Input

Input is structured in blocks. Each block starts with an integer `N` in a line, representing the number of lists that must be zipped (either 2 or 3); and `N` lists of space-separated integers ending with a single newline (note that lists may be empty). End of input is represented by a single `0`, which should not be processed.

Output

Your program must output a single line of output for each input block, contaning the ordered list in the same format as the input.

Example input & output

```
2
1 2 3 4
10 20
3
0 0 0 0 0

1 1 1 1 1
0
```

```
1 10 2 20 3 4
0 1 0 1 0 1 0 1 0 1
```

Additional guidance

You ***must*** use the provided template. Also, you cannot use *new*, *delete*, or copy node *values* around in your code (copying and/or reassigning pointers is always fine in this kind of exercises).