Exercise 8-1  Distributed counting

Write a dataflow program that performs distributed counting independently from the number of processes and workers.

The first worker (with index 0) starts counting with 1. This number is then sent to the second worker. The second worker prints the received value, increments it by one, and sends it to the next worker. The \( n \)th worker (\( n \) being the total number of workers in the dataflow program) sends its result to worker 0. The counting continues until some predefined number has been reached, then all processes terminate.

a) Write a dataflow program that prints the index of the current worker and the number of total workers to the screen. Execute this program with two processes each using two workers.

b) Extend the dataflow program so that the workers count from 1 to 10 as described above. Use method \texttt{fn loop_variable} to exert control over the numbers that are printed to the screen.

c) In the previous exercise the end of the dataflow loop is determined by the timestamp of the loop, not by the numbers moving through the dataflow network. Adapt the code such that a custom operator determines whether numbers are being forwarded to downstream operators. As soon as the operator sees the number 10 no forwarding happens and the dataflow program terminates.