—— —— ——— ———— —————————

General:

  The active element machine (AEM) is a neuromorphic machine, developed by Michael Stephen Fiske.
  My math genealogy is here: < https://www.mathgenealogy.org/id.php?id=17810 >.

  The 2011 Springer paper describes the AEM: < https://www.aemea.org/AUTN2011/AEM_msf_2011.pdf >.

  The 2012 Turing Centennial paper constructs an AEM, using quantum randomness that executes a
  Universal Turing machine with Turing Incomputable firing patterns to an external observer.
  < https://www.aemea.org/Turing100 >   The paper constructs a quantum random blackbox.

  The AEM is currently implemented as a virtual machine. The AEM is NOT a register machine,
  even though the virtual machine paradoxically executes on a register machine.

  An AEM program has element, connection, fire, and meta commands.
  Meta commands self-modify the program as it is executing.

  The virtual machine is implemented in the C programming language with a LISP (newlisp) shell that
  makes calls into dynamic library AEM.dylib. The main LISP files are AEM_run.lsp & AEM_parse.lsp.

  Some example AEM programs are shown executing at < https://www.aemea.org/AEM >.
  AEM.dylib executes an artificial time delay between each AEM command so you can watch them "execute".
  AEM commands printed sequentially at a time node simultaneously execute in an AEM hardware machine.

  At < https://www.aemea.org/AEM > below the heading "AEM K5 Example Programs" is some AEM code.
  For example, read "AEM K5 red heal source" and then click on "AEM executes K5 red heal source".

  If you want to understand what these simple K5 AEM programs are illustrating, study the paper
  < https://www.aemea.org/HICSS2023/Dynamical_Systems_that_Heal.pdf >.   See pages 3, 4, and 5.

  "Dynamical Systems that Heal" won Best Paper on January 6, 2023.
  < https://www.aemea.org/HICSS2023/HICSS56_Best_Paper.pdf >

—— —— ——— ———— —————————

Installation:

  Install newlisp on MacOS: Goto < https://www.newlisp.org >. Click on Downloads.

  AEM.dylib is compiled for x86_64 on MacOS.
  If you want an AEM dynamic library for Linux, send an email to mf@aemea.org.

—— —— ——— ———— —————————

Usage:

  Put all the ".aem" files into the same directory as "AEM.dylib" and the ".lsp" files.

  $ newlisp   AEM_run.lsp   K5_1_red_triangle.aem

  $ newlisp   AEM_run.lsp   K5_red_triangle_connection_heal.aem

  $ newlisp   AEM_run.lsp   K5_no_monochrome_triangle.aem

—— —— ——— ———— —————————

Lisp shell:

  AEM_run.lsp   "Main" newlisp AEM program. Makes functions calls into AEM.dylib

  AEM_constants.lsp   Specifies constants such as the initial AEM clock time and AEM error codes.

  AEM_parse.lsp   Preprocesses an AEM program with file suffix ".aem".

  AEM_log.lsp   Enables the logging of AEM computational activity.

  AEM_errors.lsp   Contains procedures called during the AEM execution if an error occurs.