##CLOJURE.CLhs.nt.ns.clHASKELL:PythonCODES.p used AGAINST AN ANTI-WH AND ANTI-CIA NETWORK WHICH ACCORDING TO SOURCES ATTEMPTED TO INTERRUPT A POSTIVE CIAOPS SERIES WHICH AIDED AND ABETTED THE WHIHOU\*&\_ IN SEVERAL INSTANCES IN TEXAS, FLORIDA, AND CALIFORNIA/NEW MEXICO WHICH WERE AUGEMENT\* THE NUMBER OF QUALIFIED CONGRESSWOMEN IN THOSE STATES AND REGIONS\*&/ WHO HAD ATTENDED HARVARD UNIVERSITY OR HAD ATTENDED TO SELECTING A UNIVERSITY FROM THE IVY LEAGUE PARTICULARILY BOWLING GREEN STATE UNIVERSITY AND PRINCETON. ALSO THE ANTI-CIA ANTI-WHn NETWORK ATTEMPTED TO SUBVERGE ROCKET-POWERED AERIAL-WEAPONRY FROM BEING ACQUIRED THROUGH TREATY AND CONTRACTUAL.BUSINESS FROM THE WH. THE WH SUBSEQUENTLY MADE ALTERNATIVE PLANS FOR WEAPONS-AERIAL-ACQUISTION. THE ANTI-CIA NETWORK MADE COMMENT SUCH THAT THEY WOULD CONTINE THEIR RUFFIAN WAYS AND MANNERS.

<# CLOJURE.cl//http://n.www.python.py(python.orgCLOJURE.org.RUBY.LIB:python//CLOJURESCRIPT::clojure<n>?-@#2<n>

<require::HASKELL.hs//clojure.cl>

<python CLOJUREclpyNetworkHippoBungalow.RUBY.org<n>>?

<python fbi.org ? [py.FBI/python.py//#2@2<n](mailto:py.FBI/python.py//#2@2<n) TRUE? == YES \*.org //?.8ct.c n>

## THESE B2-POUND-FACTOR SER-SEQ n TRANSDUCE CLOJURESCRIPTcopyFROM BOOLExpression=TRUE to B2 SEQnetworkHIPPO through OVER A MILLION TRANSLATIONS OF e.

def nHIPPOcopyftp{[list, list::2, list@78:port:324$#, listings2, n ]}

for {[ (concat::\*HIPPOcopy\*\* [ “ ‘iSEQnb = kw://Nutpass@2nb/http:PASSN’ “, [“ ‘324HIPPO-ndef.78Portnut\* = FALSE’ “ (n-1) = [Nutpass@::http://ftp:http::n.w.:iSEQnb::localhost::::78Portnut\*](mailto:Nutpass@::http://ftp:http::n.w.:iSEQnb::localhost::::78Portnut*) =

unless i = “i9xn” while i = 3:

[Nutpass@2=ftp::singlular.action.module.py::Python::::listings2](mailto:Nutpass@2=ftp::singlular.action.module.py::Python::::listings2) for ftp (n):

f()

return f(n)::2

f()

]

}} }}} Nutpassfn()

#CLOJURE.cl CODES WHICH TRANSLATE http:://forlistings2 to NutpassPython.pyftp:::onemillion78Port:::localhost78Portnut:::4342$\*series:

n\* random.math.seq(2\*3\*\*432-OPENSOURCEportNEXTPORT.4342$series:

defMacroHIPPOn open seq close.flush let(NEXTpORT onemillion Exitseries’’localport::Nutp\* save.Names.## Opensource::f() while n </ 3 > 0 + onemillionNextpOrt::defMacroHIPPOn nn ((merge:seq aport=localport:2 Nutpass+.\*port:::[0-9] } )))

aseq:: Totalarrayseq::: = ( merge save close flush ) }}

))))

defFastnutpst let (/y [hs, h] = 4$) [header, $4.clojurescript:::browser::Opensource::f() n-1@2) add.merge

merge.header + Totalarrayseq con aport[NextpOrt..]//.//.. merge

return (save.close.flush))) pass::totalPortnut78-2.browser.Tab1-3.flush

return f() flush.close

Nutpass = []

merge [] {[ con aport save close flush ]}}}

h <- Int <- Int -< Int [a], [b]

set! ( :h -> [] <- Int <- Int flush.close: h): con aport.$n<n>?

aarray: get

Nutpass = []