⍝ The Co-dfns Compiler

⍝ High-performance, Parallel APL Compiler

⍝ *Copyright © 2011-2017 Aaron W. Hsu* [*arcfide@sacrideo.us*](mailto:arcfide@sacrideo.us)   
⍝   
⍝ This program is free software: you can redistribute it and/or modify it under the terms of the GNU Affero  
⍝ General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your   
⍝ option) any later version.  
⍝   
⍝ This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the   
⍝ implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU Affero   
⍝ General Public License for more details.  
⍝   
⍝ You should have received a copy of the GNU Affero General Public License along with this program.   
⍝ If not, see <http://www.gnu.org/licenses/>

:Namespace codfns

⎕IO ⎕ML ⎕WX ← 0 1 3  
 VERSION ← 2017 12 0  
 AF∆PREFIX ← '/usr/local'  
 AF∆LIB ← ''  
 VS∆PS ← '\2017\'∘,¨ 'Enterprise' 'Professional' 'Community' ,¨ ⊂'\VC\Auxiliary\Build'  
 VS∆PS ,← ⊂' 14.0\VC'  
 VS∆PS ,¨⍨← ⊂'\Program Files (x86)\Microsoft Visual Studio'  
 VS∆PS ,¨← ⊂'\vcvarsall.bat'

Cmp ← { \_ ← 1 ⎕NDELETE ⍺, soext ⍬  
 \_ ← (⍎ opsys 'vsc' 'gcc' 'clang') ⍺ ⊣ (⍺, '.cpp') put⍨ gc tt ⊃ a n ← ps ⍵  
 \_ ← ⎕NEXISTS ⍺, soext ⍬ ⋄ \_ : n  
 'COMPILE ERROR' ⎕SIGNAL 22}  
 MkNS ← {ns ⊣ ⍺∘{ns.⍎⍺ mkf ⍵}¨ (1 = 1 ⌷ ⍉⍵)⌿ 0 ⌷ ⍉⍵ ⊣ ns ← #.⎕NS ⍬}  
 Fix ← {⍺ MkNS ⍺ Cmp ⍵}  
 Xml ← {⎕XML (0 ⌷ ⍉⍵), (,∘⍕⌿ 2 ↑ 1 ↓ ⍉⍵), (⊂''), ⍪(⊂(¯3 + ≢⍉⍵) ↑ ,¨ 'nrsgvyel') ,∘⍪¨ ↓ ⍕∘,¨ ⍉ 3 ↓ ⍉⍵}  
 MKA ← {mka ⊂ ⍵ ⊣ 'mka' ⎕NA 'P ', (⍺, soext ⍬), '|mkarray <PP'}  
 EXA ← {exa ⍬ ⍵ ⊣ 'exa' ⎕NA (⍺, soext ⍬), '|exarray >PP P'}  
 FREA ← {frea ⍵ ⊣ 'frea' ⎕NA (⍺, soext ⍬), '|frea P'}

opsys ← {⍵ ⊃⍨ 'Win' 'Lin' 'Mac' ⍳ ⊂ 3 ↑ ⊃ '.' ⎕WG 'APLVersion'}  
 soext ← {opsys '.dll' '.so' '.dylib'}  
 tie ← {0 :: ⎕SIGNAL ⎕EN ⋄ 22 :: ⍵ ⎕NCREATE 0 ⋄ 0 ⎕NRESIZE ⍵ ⎕NTIE 0}  
 put ← {s ← (¯128 + 256 | 128 + 'UTF-8' ⎕UCS ⍺) ⎕NAPPEND (t ← tie ⍵) 83 ⋄ 1 : r ← s ⊣ ⎕NUNTIE t}  
 mkf ← { fn ← (⍺, soext ⍬), '|', ('∆' ⎕R '\_\_' ⊢ ⍵), '\_dwa '  
 f ← ⍵, '←{\_←''dya''⎕NA''', fn, '>PP <PP <PP'' ⋄ '  
 f ,← '\_←''mon''⎕NA''', fn, '>PP P <PP'' ⋄ '  
 f, '0=⎕NC''⍺'':mon 0 0 ⍵ ⋄ dya 0 ⍺ ⍵} ⋄ 0'}

ccf ← {' -o ''', ⍵, '.', ⍺⍺, ''' ''', ⍵, '.cpp'' -laf', AF∆LIB, ' > ', ⍵, '.log 2>&1'}  
 cci ← {'-I''', AF∆PREFIX, '/include'' -L''', AF∆PREFIX, '/lib'' '}  
 cco ← '-std=c++11 -Ofast -g -Wall -fPIC -shared '  
 ucc ← {⍵⍵ (⎕SH ⍺⍺, ' ', cco, cci, ccf ) ⍵}  
 gcc ← 'g++' ucc 'so'  
 clang ← 'clang++' ucc 'dylib'  
 vsco ← { z ← '/W3 /wd4102 /wd4275 /Gm- /O2 /Zc:inline /Zi /Fd"', ⍵, '.pdb" '  
 z ,← '/errorReport:prompt /WX- /MD /EHsc /nologo '  
 z, '/I"%AF\_PATH%\include" /D "NOMINMAX" /D "AF\_DEBUG" '}  
 vslo ← { z ← '/link /DLL /OPT:REF /INCREMENTAL:NO /SUBSYSTEM:WINDOWS '  
 z ,← '/LIBPATH:"%AF\_PATH%\lib" /DYNAMICBASE "af', AF∆LIB, '.lib" '  
 z, '/OPT:ICF /ERRORREPORT:PROMPT /TLBID:1 '}  
 vsc0 ← {~∨⌿ b ← ⎕NEXISTS¨ VS∆PS : 'MISSING VISUAL C++' ⎕SIGNAL 99 ⋄ '""', '" amd64' ,⍨ ⊃ b ⌿ VS∆PS}  
 vsc1 ← {' && cd "', (⊃ ⎕CMD 'echo %CD%'), '" && cl ', (vsco ⍵), '/fast "', ⍵, '.cpp" '}  
 vsc2 ← {(vslo ⍵), '/OUT:"', ⍵, '.dll" > "', ⍵, '.log""'}  
 vsc ← {⎕CMD ('%comspec% /C ', vsc0, vsc1, vsc2) ⍵}

get ← {⍺⍺ ⌷ ⍉⍵}  
 wrap ← ⍪∘(⍉ (1 + 1 ↑ ⍉) ⍪ 1 ↓ ⍉)  
 bind ← {n \_ e ← ⍵ ⋄ (0 n\_ ⌷ e) ← ⊂n ⋄ e}  
 at ← {⍺ ← ⊢ ⋄ A ⊣ ((,B) ⌿ (r A) ⍴ A) ← ⍺ ⍺⍺ (,B) ⌿ ((r ← (≢⍴ B ← ⍵⍵ ⍵) ((×/ ↑) , ↓) ⍴) A) ⍴ (A ← ⍵)}

d\_ t\_ k\_ n\_ r\_ s\_ g\_ v\_ y\_ e\_ l\_ ← ⍳ 7 + f∆ ← 4  
 d ← d\_ get ⋄ t ← t\_ get ⋄ k ← k\_ get ⋄ n ← n\_ get ⋄ r ← r\_ get ⋄ s ← s\_ get   
 g ← g\_ get ⋄ v ← v\_ get ⋄ y ← y\_ get ⋄ e ← e\_ get ⋄ l ← l\_ get

new ← {⍉⍪ f∆ ↑ 0 ⍺, ⍵} ⋄ msk ← {(t ⍵) ∊ ⊂⍺⍺} ⋄ sel ← {(⍺⍺ msk ⍵)⌿ ⍵}  
 A ← {('A' new ⍺⍺) wrap ⊃⍪/ ⍵} ⋄ Am ← 'A' msk ⋄ As ← 'A' sel  
 E ← {('E' new ⍺⍺) wrap ⊃⍪/ ⍵} ⋄ Em ← 'E' msk ⋄ Es ← 'E' sel  
 F ← {('F' new ⍺⍺) wrap ⊃⍪/ (⊂ 0 f∆ ⍴ ⍬), ⍵} ⋄ Fm ← 'F' msk ⋄ Fs ← 'F' sel  
 G ← {('G' new 0) wrap ⊃⍪/ ⍵} ⋄ Gm ← 'G' msk ⋄ Gs ← 'G' sel  
 L ← {('L' new 0) wrap ⊃⍪/ ⍵} ⋄ Lm ← 'L' msk ⋄ Ls ← 'L' sel  
 M ← {('M' new 0 '') wrap ⊃⍪/ (⊂ 0 f∆ ⍴ ⍬), ⍵} ⋄ Mm ← 'M' msk ⋄ Ms ← 'M' sel  
 N ← {'N' new 0 (⍎⍵)} ⋄ Nm ← 'N' msk ⋄ Ns ← 'N' sel  
 O ← {('O' new ⍺⍺) wrap ⊃⍪/ ⍵} ⋄ Om ← 'O' msk ⋄ Os ← 'O' sel  
 P ← {'P' new 0 ⍵} ⋄ Pm ← 'P' msk ⋄ Ps ← 'P' sel  
 S ← {'S' new 0 ⍵} ⋄ Sm ← 'S' msk ⋄ Ss ← 'S' sel  
 V ← {'V' new ⍺⍺ ⍵} ⋄ Vm ← 'V' msk ⋄ Vs ← 'V' sel  
 Y ← {'Y' new 0 ⍵} ⋄ Ym ← 'Y' msk ⋄ Ys ← 'Y' sel  
 Z ← {'Z' new 1 ⍵} ⋄ Zm ← 'Z' msk ⋄ Zs ← 'Z' sel

\_o ← {0 ≥ ⊃ c a e r ← p ← ⍺ ⍺⍺ ⍵ : p ⋄ 0 ≥ ⊃ c a e r2 ← p ← ⍺ ⍵⍵ ⍵ : p ⋄ c a e (r ↑⍨ - ⌊/ ≢¨ r r2)}  
 \_s ← {0 < ⊃ c a e r ← p ← ⍺ ⍺⍺ ⍵ : p ⋄ 0 < ⊃ c2 a2 e r ← p ← e ⍵⍵ r : p ⋄ (c ⌈ c2)(a, a2) e r}  
 \_noenv ← {0 < ⊃ c a e r ← p ← ⍺ ⍺⍺ ⍵ : p ⋄ c a ⍺ r}  
 \_env ← {0 < ⊃ c a e r ← p ← ⍺ ⍺⍺ ⍵ : p ⋄ c a (e ⍵⍵ a) r}  
 \_then ← {0 < ⊃ c a e r ← p ← ⍺ ⍺⍺ ⍵ : p ⋄ 0 < ⊃ c a e \_ ← p ← e (⍵⍵ \_s eot) a : p ⋄ c a e r}  
 \_not ← {0 < ⊃ c a e r ← ⍺ ⍺⍺ ⍵ : 0 a ⍺ ⍵ ⋄ 2 a ⍺ ⍵}  
 \_as ← {0 < ⊃ c a e r ← ⍺ ⍺⍺ ⍵ : c a e r ⋄ c (,⊂⍵⍵ a) e r}  
 \_t ← {0 < ⊃ c a e r ← ⍺ ⍺⍺ ⍵ : c a e r ⋄ e ⍵⍵ a : c a e r ⋄ 2 ⍬ ⍺ ⍵}  
 \_ign ← {c a e r ← ⍺ ⍺⍺ ⍵ ⋄ c ⍬ e r}  
 \_peek ← {0 < p ← ⊃ ⍺ ⍺⍺ ⍵ : p ⋄ 0 ⍬ ⍺ ⍵}  
 \_yes ← {0 ⍬ ⍺ ⍵}  
 \_opt ← {⍺ (⍺⍺ \_o \_yes) ⍵}  
 \_any ← {⍺ (⍺⍺ \_s ∇ \_o \_yes) ⍵}  
 \_some ← {⍺ (⍺⍺ \_s (⍺⍺ \_any)) ⍵}  
 \_set ← {(0 ≠ ≢⍵) ∧ (⊃⍵) ∊ ⍺⍺ : 0 (,⊃⍵) ⍺ (1 ↓ ⍵) ⋄ 2 ⍬ ⍺ ⍵}  
 \_tk ← {((≢,⍺⍺) ↑ ⍵) ≡,⍺⍺ : 0 (⊂,⍺⍺) ⍺ ((≢,⍺⍺) ↓ ⍵) ⋄ 2 ⍬ ⍺ ⍵}  
 \_eat ← {0 = ≢⍵ : 2 ⍬ ⍺ ⍵ ⋄ 0 (⍺⍺ ↑ ⍵) ⍺ (⍺⍺ ↓ ⍵)}

ws ← (' ', ⎕UCS 9) \_set  
 aws ← ws \_any \_ign  
 awslf ← (⎕UCS 10 13) \_set \_o ws \_any \_ign  
 gets ← aws \_s ('←' \_tk) \_s aws \_ign  
 him ← '¯' \_set  
 dot ← '.' \_set  
 jot ← '∘' \_set  
 lbrc ← aws \_s ('{' \_set) \_s aws  
 rbrc ← aws \_s ('}' \_set) \_s aws  
 lpar ← aws \_s ('(' \_tk) \_s aws \_ign  
 rpar ← aws \_s (')' \_tk) \_s aws \_ign  
 lbrk ← aws \_s ('[' \_tk) \_s aws \_ign  
 rbrk ← aws \_s (']' \_tk) \_s aws \_ign  
 semi ← aws \_s (';' \_tk \_as ('a' V∘,∘⊃)) \_s aws  
 grd ← aws \_s (':' \_tk) \_s aws \_ign  
 egrd ← aws \_s ('::' \_tk) \_s aws \_ign  
 alpha ← 'ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz∆' \_set  
 digits ← '0123456789' \_set  
 prim ← (prims ← '+-÷×|\*⍟⌈⌊!<≤=≠≥>∧∨⍲⍱⌷?⍴,⍪⌽⊖⍉∊⍷⊃⍳○~≡≢⊢⊣/⌿\⍀⊤⊥↑↓∪∩⍋⍒∇⌹') \_set  
 mop ← '¨/⌿⍀\⍨' \_set  
 dop1 ← '.⍣∘' \_set  
 dop2 ←'⍤⍣∘' \_set  
 dop3 ←'∘' \_set  
 eot ← aws \_s {'' ≡ ⍵ : 0 ⍬ ⍺ '' ⋄ 2 ⍬ ⍺ ⍵} \_ign  
 digs ← digits \_some  
 odigs ← digits \_any  
 int ← aws \_s digs \_s (him \_opt) \_s aws  
 float ← aws \_s (odigs \_s dot \_s int \_o (digs \_s dot)) \_s aws  
 name ← aws \_s (alpha \_o (digits \_some \_s alpha) \_some) \_s aws  
 aw ← aws \_s ('⍺⍵' \_set) \_s aws  
 aaww ← aws \_s (('⍺⍺' \_tk) \_o ('⍵⍵' \_tk)) \_s aws  
 sep ← aws \_s (('⋄', ⎕UCS 10 13) \_set \_ign) \_s aws  
 nss ← awslf \_s (':Namespace' \_tk) \_s aws \_s (name \_opt) \_s awslf \_ign  
 nse ← awslf \_s (':EndNamespace' \_tk) \_s awslf \_ign

Sfn ← aws \_s (('TFF⎕' \_tk) \_o ('TFFI⎕' \_tk)) \_s aws \_as {P ⌽∊⍵}  
 Prim ← prim \_as P  
 Vt ← {((0 ⌷ ⍉⍺) ⍳ ⍵) 1 ⌷ ⍺ ⍪ '' ¯1}  
 Var ← {⍺ (aaww \_o aw \_o (name \_as ⌽) \_t (⍺⍺ = Vt) \_as (⍵⍵ V∘,∘⊃)) ⍵}  
 Num ← float \_o int \_as (N∘⌽)  
 Strand ← 0 Var 'a' \_s (0 Var 'a' \_some) \_as ('s' A∘⌽)  
 Pex ← {⍺ (rpar \_s Ex \_s lpar) ⍵}  
 Atom ← Strand \_o (0 Var 'a' \_as ('v'A)) \_o (Num \_some \_as ('n' A∘⌽)) \_o Pex  
 Brk ← rbrk \_s {⍺ (Ex \_opt \_s (semi \_s (Ex \_opt) \_any)) ⍵} \_s lbrk \_as ('i' E∘⌽)  
 Idx ← Brk \_s (\_yes \_as {P ,'['}) \_s Atom \_as (2 E∘⌽)  
 Blrp ← {⍺ (⍺⍺ \_s (⍵⍵ Slrp ∇)) ⍵}  
 Slrp ← {⍺ (⍺⍺ \_o (⍵⍵ \_s ∇) \_o ((1 \_eat) \_s ∇)) ⍵}  
 Fa ← { e ← ('⍵⍵' '⍺⍺', '⍺⍵') ,∘⍪⍤1 ⍉ ¯1 + 3 3 2 2 ⊤ (6 4 4 ⌿ 1 5 9) + 2 × ⍳ 14  
 a ← e (⍺{⍵ Gex \_o Ex \_o Fex Stmts \_then Fn⍨ ⍺⍺ ⍪ ⍺})⍤2 1 ⊢ ⍵  
 m ← (0 = 0 ⌷ ⍉a) ∧ ∧⌿ (∨⍀ ∘.=⍨ ⍳14) ∨ ∘.≢⍨ 1 ⌷ ⍉a  
 ~∨⌿ m : (⌈⌿ 0 ⌷ ⍉a) ⍬ ⍺ ⍵  
 (1 = +⌿ m) ∧ 2 > m ⍳ 1 : 0 (,⊂('F' new 1) wrap ⊃⊃ m ⌿ 1 ⌷ ⍉a) ⍺ ⍵  
 z ← ('F' new 'a') wrap ⊃ (m ⌿ 'F' new¨ 1 + ⍳14) ⍪.(wrap∘⊃) m ⌿ 1 ⌷ ⍉a  
 0 (,⊂z) ⍺ ⍵}  
 Fn ← { ns ← n z ⌿⍨ m ← {(Fm ⍵) ∧ ¯1 ∊⍨ k ⍵} z ← ⊃⍪/ ⍵ ⋄ 0 = ≢ns : 0 (,⊂z) ⍺ ''  
 p ← ⍺∘Fa¨ ns ⋄ 0 < c ← ⌈/ ⊃¨p : c ⍬ ⍺ ⍵  
 0 (⍪/ (⊂ 0 4 ⍴ ⍬) ,⍨ p {⍵ ((d +∘⊃ ⊣), 1 ↓⍤1 ⊢) ⊃⍪/ 1 ⊃ ⍺}¨at{m} ↓z) ⍺ ''}  
 Pfe ← {⍺ (rpar \_s Fex \_s lpar) ⍵}  
 Bfn ← rbrc Blrp lbrc \_as ('F' new ¯1 ,∘⊂∘⌽ 1 ↓ ¯1 ↓ ⊢)  
 Fnp ← Prim \_o (1 Var 'f') \_o Sfn \_o Bfn \_o Pfe  
 Mop ← {⍺ ((mop \_as P) \_s Afx \_as (1 O)) ⍵}  
 Dop1 ← {⍺ ((dop1 \_as P) \_s Afx \_as (2 O∘⌽)) ⍵}  
 Dop2 ← {⍺ (Atom \_s (dop2 \_as P) \_s Afx \_as (2 O∘⌽)) ⍵}  
 Dop3 ← (dop3 \_as P) \_s Atom \_as (2 O∘⌽) \_o (dot \_s jot \_as (P∘⌽) \_as (1 O))  
 Bop ← {⍺ (rbrk \_s Ex \_s lbrk \_s (\_yes \_as {P ,'['}) \_s Afx \_as (2 O∘⌽)) ⍵}  
 Afx ← Mop \_o (Fnp \_s (Dop1 \_o Dop3 \_opt) \_as (⊃wrap/∘⌽)) \_o Dop2 \_o Bop  
 Trn ← {⍺ (Afx \_s ((Afx \_o Idx \_o Atom) \_s (∇ \_opt) \_opt)) ⍵} \_as ('t' F∘⌽)  
 Bind ← {⍺ (gets \_s (name \_as ⌽) \_env (⍺⍺{(⊃⌽⍵) ⍺⍺ ⍪ ⍺}) \_as (⍵⍵ new 'b', ⊢)) ⍵}  
 Asgn ← gets \_s Brk \_s (name \_as ⌽ \_t (0 = Vt) \_as ('a' V∘,∘⊃)) \_as ('a' E∘⌽)  
 Fex ← Afx \_s (Trn \_opt) \_s (1 Bind 'F' \_any) \_as (⊃wrap/∘⌽)  
 App ← Afx \_s (Idx \_o Atom \_s (dop2 \_not) \_opt) \_as {(≢⍵) E ⌽⍵}  
 Ex ← Idx \_o Atom \_s {⍺ (0 Bind 'E' \_o Asgn \_o App \_s ∇ \_opt) ⍵} \_as (⊃wrap/∘⌽)  
 Gex ← Ex \_s grd \_s Ex \_as (G∘⌽)  
 Nlrp ← sep \_o eot Slrp (lbrc Blrp rbrc)  
 Stmts ← {⍺ (sep \_any \_s (Nlrp \_then (⍺⍺ \_s eot∘⌽)) \_any \_s eot) ⍵}  
 Ns ← nss Blrp nse \_then (Ex \_o Fex Stmts \_then Fn) \_s eot \_as M

ps ← {0 ≠ ⊃ c a e r ← (0 2 ⍴ ⍬) Ns ∊{⍵ /⍨ ∧\ '⍝' ≠ ⍵}¨⍵ ,¨ ⎕UCS 10 : ⎕SIGNAL c ⋄ (⊃a) e}

scp ← (+⍀ Fm) ⊢∘⊂⌸ ⊢  
 prf ← ((≢ ↑ ¯1 ↓ ⊢ (⌿⍨) 0 ≠ ⊢)⍤1 ↑∘r) ⊢  
 blg ← {⍺ ← ⊢ ⋄ ⍺((prf (⌈/ (⍳∘≢⊢) ×⍤1 (1 ↓ ⊣) ∧.(= ∨ 0 = ⊢)∘⍉ ⊢) ⍺⍺ (⌿∘↑) r)⌷⍤0 2 ⍵⍵ (⌿⍨) ⍺⍺)⍵}  
 enc ← ⊂ ⊣ ,∘⊃ ((⊣, '\_', ⊢)/ (⊂''), (⍕¨ ⊢ (/⍨) 0 ≠ ⊢))  
 veo ← ∪ ((⊂'%u'), (,¨prims), ⊣) ~⍨∘{⊃,/ {⊂⍣(1 ≡ ≡⍵) ⊢ ⍵}¨⍵} ¯1 ↓ ⊢ (/⍨) (∧/¨ 0 ≠ ((⊃ 0 ⍴ ⊢)¨⊢))  
 ndo ← {⍺ ← ⊢ ⋄ m ⊃∘(⊂, ⊢)¨ ⍺∘⍺⍺¨¨ ⍵ ⊃∘(,∘⊂⍨⊂)¨⍨ m ← 1 ≥ ≡¨⍵}  
 n2f ← (⊃,/) ((1 = ≡) ⊃ ,∘⊂⍨∘⊂)¨

rn ← ⊢ ,∘↓ (1 + d) ↑⍤¯1 (+⍀ d ∘.=∘⍳ 1 + (⌈/ 0, d))  
 rd ← ⊢ , (+/ ↑∘r ∧.(= ∨ 0 = ⊢)∘⍉∘↑∘r ⊢ (⌿⍨) Fm ∧ 1 ∊⍨ k)  
 df ← ⊢ (⌿⍨) (+\ 1 = d) (~ ⊣ ∊ ⊣ (/⍨) (1 = d) ∧ (~ 'b' ∊⍨ k) ∧ Om ∨ Fm) ⊢  
 dua ← ((~Gm) ∧ Fm ∨ ↓∘prf ∊ r∘Fs) (⊣ (⍀∘⊢) (d (⌿⍨) ⊣) (0, 1 ↓ (¯1 ⌽ ⊢) ∧ ⊣ = ¯1 ⌽ ⊣) ⊣ (⌿∘⊢) 0 ∊⍨ n) ⊢  
 du ← ⊢ (⌿⍨∘~) dua ∨∘(∨/) (prf ∧.(= ∨ 0 = ⊢)∘⍉ prf (⌿⍨) dua) ∧ ↑∘r ∧.≥∘⍉ dua (⌿∘⊢) ↑∘r × 0 = prf  
 lfh ← (0 ≠ 1 ⌷ ⊣) ⊃ (⊂∘⍉∘⍪ 0 'M' 0 '', 0 ,⍨ (⊂⊣)) ,∘⊂∘⍉∘⍪ 1 'F' 1, ('fn' enc ⊣), (⊂⊣), 5 ↓∘, 1 ↑ ⊢  
 lfn ← (d, 'Of', 3 ↓ ⊢)⍤1 at (Fm ∧ 'b' ∊⍨ k) (d, 'Vf', ('fn' enc∘⊃ r), 4 ↓ ⊢)⍤1 at (Fm ∧ 1 ∊⍨ k)  
 lf ← (⊃⍪/) (1, 1 ↓ Fm ∧ 1 ∊⍨ k) blg (↑r) (⊂ lfh ⍪∘(((⊢ - (⊃ - 2 ⌊ ⊃)) d), 1 ↓⍤1 ⊢) lfn)⌸ 1 ↓ ⊢  
 dn ← ((0 ∊⍨ n) ∧ (Am ∧ 'v' ∊⍨ k) ∨ Om ∧ 'f' ∊⍨ k) ((~⊣) (⌿∘⊢) (d - ¯1 ⌽ ⊣), 1 ↓[1] ⊢) ⊢  
 mrep ← (1 + ⊃), 'P' 0 (,'⊢'), (⊂'') ,⍨ ¯1 ↓ 4 ↓∘, 1 ↑ ⊢  
 mreu ← ⊃, 'E' 'u', (⊂'') ,⍨ ¯1 ↓ 3 ↓∘, 1 ↑ ⊢  
 mre ← (⊃⍪/) (-∘⊃ Vm ∨ Am)∘⊃∘⌽ (↓, (((⊢ ⍴⍨ (≢⍉) ,⍨ ≢ × 2 < ≢) mreu ⍪ mrep ⍪ (1 + d), 1 ↓⍤1 ⊢)¨ ↑)) ⊢  
 mrs ← ⊢ ⊂[0]⍨ 1, 1 ↓ d = 1 +∘⊃ d  
 mrk ← (-∘(+/ ∧\)∘⌽ Lm) (↑ ⍪⍨∘(mre (mre mrs)¨ at (Gm∘(⊃⍪/) 1 ↑¨ ⊢)∘mrs) ↓) ⊢  
 mr ← (⊃⍪/) ((1 ↑ ⊢), (mrk¨ 1 ↓ ⊢))∘scp  
 ur ← ((2 ↑ ⊢), 1, ('um' enc∘⊃ r), 4 ↓ ⊢)⍤1 at (Em ∧ 'u' ∊⍨ k)  
 rt ← ⊢, (∨\ Fm) + (+⌿ prf ∧.(= ∨ 0 = ⊣)∘⍉⍨∘↑∘r Ms ⍪ Gs) - Fm  
 nm ← ((3 ↑ ⊢), ('fe' enc∘⊃ r), 4 ↓ ⊢)⍤1 at ((0 ∊⍨ n) ∧ Em ∨ Om ∨ Am)  
 lgg ← (⍪/ 1 ↓ ⊢) ⍪∘⊃⍨ ⊣ (((¯1 + d), 2 ,⍨ t, k, n, r ,∘⍪ s) ⍪ ⊣ ⍪ 3, 'V', 'a', 3 (↓⍤1) 1 ↑ ⊢)∘⊃ 1 ↑ ⊢  
 lg ← (⊃⍪/) ⊢ ((⊂ ⊣ (⌿⍨∘~) (∨\ ⊢)), (((1 ↑ ⊢) lgg ⊢ ⊂[0]⍨ d = 1 + ⊃)¨ ⊂[0]⍨)) Gm ∧ 1 ⌽ Em  
 fet ← (d, 'V' 0, 3 ↓ ⊢)⍤1 at (0, 1 ↓ Em ∨ Om ∨ Am) (d, 'Av', 3 ↓ ⊢)⍤1 at (Em ∧ 'b' ∊⍨ k)  
 fee ← (⍪/⌽) (Mm ∨ Em ∨ Om ∨ Am) blg ⊢ ((⊃∘⌽ ⊢) (⊂ (d - -⍨∘ ⊃), 1 ↓⍤1 ⊢)∘fet ⊣ ⍪ ¯1 ↓⍤1 ⊢)⌸ ⊃ ,⍨ 1 ↓ ⊢  
 fe ← (⊃⍪/) (+⍀ d ≤ g) (⊂ (⊢ ↑⍨ 1 =∘≢ ⊢) ⍪∘⊃∘fee ⊢)⌸ ⊢  
 can ← (+\ Am ∨ Om) ((, 1 ↑ ⊢) ,∘(⊂ (¯1 + 2 ⌊ ≢) ⊃ (⊂∘⊂ ⊃), ⊂)∘n 1 ↓ ⊢)⌸ ⊢  
 cas ← (¯1 ⌽ (Am ∨ Om) ∧ 'vf' ∊⍨ k) ∨ (↓prf) ∊∘r ⊢ (⌿⍨) Am ∧ 'n' ∊⍨ k  
 ca ← (can ⊢ (⌿⍨) cas ∨ Am ∨ Om ∧ 'f' ∊⍨ k) ⊣ at (Am ∨ Om ∧ 'f' ∊⍨ k) ⍬ ,∘⊂⍨ ⊢ (⌿⍨∘~) cas  
 lj ← (⊃⍪/) (1 ↑ scp), ((⊢ ⍪ 2 'L' 0 0, 2 '' ,⍨ ¯2 ↓ 4 ↓∘, 1 ↑ ⊢)¨ 1 ↓ scp)  
 sd ← (⊃⍪/) (1 ↑ scp), (n Fs) (d, 'Vf', (⊂⊣), 4 ↓ ⊢)⍤1 at ((⊂, '∇') ∊⍨ n)¨ 1 ↓ scp  
 inm ← ∨⌿ ¯1 (⌽ ∨ ⊢) 1 2 (⌽ ∨ ⊢) (¯1 ¯2 ⌽ Em ∧[1] 1 2 ∘.= k) ∧⍤1 Vm ∧ n ∊∘n Fs  
 inp ← (Em ∧ ⊣) ∨ 1, 2 ≠/ ⊣  
 inza ← (1 ↑ 1 ↓ ⊣) (⌿⍨∘≢) at ((⊂, '⍺') ∊⍨ n) (¯1 ↑ ⊣) (⌿⍨∘≢) at ((⊂, '⍵') ∊⍨ n) ⊢  
 inz ← (1 ↑ ⊣) (d, t, k, 3 ↓⍤1 (⌿⍨∘≢)) at (0 ,⍨ 2 ≠/∘⌽ (∨\∘⌽ Em)) inza  
 inn ← (3 ↑⍤1 ⊢), ((⊣ ⍴¨⍨ 1 + 0 ⌈ (⌈/∘n Gs)) (('fe' ≡ 2 ↑ ⊢) ⊃ (⊂⊢) ,∘⊂ 'fe', (⍕⊣), 2 ↓ ⊢)¨ n), (4 ↓⍤1 ⊢)  
 ins ← ⊣ (d, t, k, ((1000 × 1 + ⊣) + 1 + n + (⌈/n)), 4 ↓⍤1 ⊢) at (Lm ∨ Gm) inn  
 inr ← 1 ,∘⍪ ⊢ inz¨ (⍳∘≢ ⊢) ins¨ ((⊃∘n¨ ⊣) ⍳ ((⊃ n (⌿⍨) Vm ∧ 'f' ∊⍨ k)¨ ⊢)) ⊃¨ (⊂ 1 ↓¨ ⊣) ,∘⊂¨ ⊢  
 in ← (⊃⍪/)∘(⊢/) (1 ↓ scp) inr∘((0 ⍴ ⊂ 0 8 ⍴ 0), ⊢/) at (⊣/) inm ((⊃¨ inp ⊂ Em ∧ ⊣) ,∘⍪ inp ⊂[0] ⊢) ⊢

pcc ← (⊂ ⊢ (⌿⍨) Am ∨ Om ∧ 'f' ∊⍨ k)∘((⍳∘∪⍨ n) ⌷⍤0 2 (1 ⌈ ≢) ↑ ⊢)∘(⊃⍪⌿)∘⌽ (⌿∘⊢)  
 pcb ← ((, ∧.(= ∨ 0 = ⊣)∘⍪)⍤2 1⍨∘↑∘r Ms ⍪ Fs) pcc⍤1 ((⊢ (⌿⍨) (d = g) ∧ Am ∨ Em ∨ Om)¨ scp)  
 pcv ← (d, 'V', ('af' ⊃¨∘⊂⍨ Om), (⊃¨v), r, s, (⊂⍬) ,⍨∘⍪ g) at (Om ∨ Am ∧ 'v' ∊⍨ k)  
 pc ← (⊃⍪/) pcb {(pcv d (⊣, 1 ↓⍤1 ⊢)(⍺ ↑⍨ 1 ⌈ ≢⍺) ⌷⍤0 2⍨ (n ⍺) ⍳ n) at (Vm ∧ (n ⍺) ∊⍨ n) ⍵}¨ scp  
 da ← ⊢ (⌿⍨∘~) (Am ∧ d = g) ∨ (0 ,⍨ 2 ∧/ Lm) ∨ (Lm ∧ ¯1 ⌽ Am ∧ d = g) ∨ Om ∧ ('f' ∊⍨ k) ∧ 1 ≠ d  
 fce ← (⊃∘n Ps) {⊂⍎ ' ⍵' ,⍨ (≢⍵) ⊃ '' (⍺, '⊃') ('⊃', ⍺, '/')} (v As)  
 fcm ← (∧/ Em ∨ Am ∨ Pm) ∧∘~ 'ui' ∊⍨∘⊃∘⊃ k  
 fc ← ((⊃⍪/) (((d, 'An', 3 ↓ ¯1 ↓ ,) 1 ↑ ⊢), fce)¨ at (fcm¨)) ('MFOEL' ∊⍨ t) ⊂[0] ⊢  
 ce ← (+\ Fm ∨ Gm ∨ Em ∨ Om ∨ Lm) ((¯1 ↓∘, 1 ↑ ⊢) ,∘⊂ (⊃∘v 1 ↑ ⊢) ,∘(Am ⊃¨∘↓ n ,∘⍪∘n2f v) 1 ↓ ⊢)⌸ ⊢  
 ll ← (⊢ (⌿⍨) 1 ⌽ Lm) (((⊂⊂'%l') ,∘⊂¨∘n ⊣) ,⍨ ¯1 ↓⍤1 ⊢) at Lm ⊢  
 fv ← (⊃⍪/) (((1 ↓ ⊢) ⍪⍨ (, 1 7 ↑ ⊢) ,∘⊂∘n ¯1 ↑ ⊢)¨ scp)  
 nv ← (¯1 ↓⍤1 ⊢), (¯1 ⊖ ≢ ⊃ ⊢ ,∘⊂⍨ (⊂'%u' '%f' '%u'), (⊂ '%u' '%i', ⊢), (⊂ (⊂'%u'), ⊢))¨∘v  
 lt ← (⊂⍬) ,⍨ ⊢  
 val ← (n ⍳∘∪ n) ,¨ ⊢ (⊢ + (≢⊣) × 0 = ⊢) (⌈/ (⍳≢) ×⍤1 (∪n) ∘.((⊂⊣) ∊ ⊢) (n2f¨ v))  
 vag ← ∧∘~∘(∘.=⍨∘⍳ ≢)⍨ (∘.(((1 ⌷ ⊢) > 0 ⌷ ⊣) ∧ (0 ⌷ ⊢) < 1 ⌷ ⊣)⍨ val)  
 vae ← (∪n) (⊣ ,⍤0 ⊣ (⌷⍨⍤1 0)∘⊃ ((⊢ ,∘⊃ (⍳∘≢ ⊣) ~ ⊢ (⌿⍨) (≢⊢) ↑ ⊣)/∘⌽ (⊂⍬) ,∘↓ ⊢)) vag  
 vac ← (((0 ⌷∘⍉ ⊣) ⍳∘⊂ ⊢) ⊃ (1 ⌷∘⍉ ⊣) ,∘⊂ ⊢) ndo  
 va ← ((⊃⍪/) (1 ↑ ⊢), (((vae Es) (d, t, k, (⊣ vac n), r, s, g, y ,∘⍪⍨ (⊂⊣) vac¨ v) ⊢)¨ 1 ↓ ⊢)) scp  
 avb ← {(((,¨'⍺⍵') ↑⍨ 1 ↓ ⍴) ⍪ ⊢) ⍺ ⌷⍨⍤2 0 ⊢ ⍺⍺ ⍳ ⍺⍺ ∩⍨ (↓ (⌽ 1 +∘⍳ 0 ⍳⍨ ⊢) ((≢⊢) ↑ ↑)⍤0 1 ⊢) ⊃ r ⍵}  
 avi ← ¯1 0 + (⍴⊣) ⊤ (,⊣) ⍳ (⊂⊢)  
 avh ← {⊂ ⍵, (n⍵) ((⍺⍺ (⍵⍵ avb) ⍵) {⍺⍺ avi ndo (⊂⍺), ⍵})¨ v ⍵}  
 av ← (⊃⍪/) (+\ Fm) {⍺((⍺ ((∪∘⌽ (0 ⍴ ⊂''), n) Es)⌸ ⍵) avh (r (1 ↑ ⍵) ⍪ Fs ⍵))⌸ ⍵} ⊢  
 rlf ← (⌽ ↓ (((1 ⊃ ⊣) ∪ ⊢ ~ 0 ⌷ ⊣)/∘⌽ (⊂⍬), ↑)⍤0 1⍨ 1 +∘⍳ ≢) (⊖ 1 ⊖ n ,⍤0 (⊂⊣) veo¨ v)  
 rl ← ⊢ ,∘(⊃,/) (⊂∘n Os ⍪ Fs) rlf¨ scp  
 vc ← (⊃⍪/) (((1 ↓ ⊢) ⍪⍨ (1 7 ↑ ⊢), (≢∘∪∘n Es), 1 ¯3 ↑ ⊢)¨ scp)  
 eff ← (⊃⍪/) ⊢ (((⊂∘⍉∘⍪ d, 'Fe', 3 ↓ ,) 1 ↑ ⊣), 1 ↓ ⊢) (d =∘⊃ d) ⊂[0] ⊢  
 ef ← (Fm ∧ ¯1 =∘×∘⊃¨ y) ((⊃⍪/) (⊂ ⊢ (⌿⍨)∘~ (∨\ ⊣)), (eff¨ ⊂[0])) ⊢  
 ifn ← 1 'F' 0 'Init' ⍬ 0 1, (4 ⍴ 0) ⍬ ⍬ ,⍨ ⊢  
 if ← (1 ↑ ⊢) ⍪ (⊢ (⌿⍨) Om ∧ 1 = d) ⍪ ((⊢ wrap⍨∘ifn∘≢∘∪ n) ⊢ (⌿⍨) Em ∧ 1 = d) ⍪ (∨\ Fm) (⌿∘⊢) ⊢  
 fgz ← (1 ↑ ⊢) ⍪ (((¯1 + d), 1 ↓⍤1 ⊢) 1 ↓ ⊢) ⍪ 2, 'G', 1, 3 ↓⍤1 (¯1 ↑ ¯1 ↓⍤1 ⊢) ,∘n 1 ↑ ⊢  
 fg ← (⊃⍪/) (fgz¨ at (Gm∘(⊃⍪/) 1 ↑¨ ⊢) ⊢ ⊂[0]⍨ d = 2 ⌊ g)  
 fft ← (, 1 ↑ ⊢) (1 'Z', (2 ↓ ¯5 ↓ ⊣), (v⊣), n, y, (⊂ 2 ↑∘,∘⊃∘⊃ e), l) (¯1 ↑ Es)  
 ff ← ((⊃⍪/) (1 ↑ ⊢), (((1 ↑ ⊢) ⍪ (((¯1 + d), 1 ↓⍤1 ⊢) 1 ↓ ⊢) ⍪ fft)¨ 1 ↓ ⊢)) scp  
 fzh ← ((∪n) ∩ (⊃∘l ⊣)) (¯1 ⌽ (⊂⊣), ((≢⊢) - 1 + (⌽n) ⍳ ⊣) ((⊂ ⊣ ⊃¨∘⊂ (⊃¨e)), (⊂ ⊣ ⊃¨∘⊂ (⊃¨y)) ,∘⊂ ⊣) ⊢) ⊢  
 fzf ← 0 ≠ (≢∘⍴¨∘⊃∘v ⊣)  
 fzb ← (((⊃∘v ⊣) (⌿⍨) fzf), n) ,∘⍪ ('f'∘,∘⍕¨∘⍳ (+/fzf)), ('s'∘,∘⍕¨∘⍳∘≢ ⊢)  
 fzv ← ((⊂⊣) (⊖↑)⍨¨ (≢⊣) (- +∘⍳ ⊢) (≢⊢)) ((⊢ ,⍨ 1 ⌷∘⍉ ⊣) ⌷⍨ (0 ⌷∘⍉ ⊣) ⍳ ⊢)⍤2 0¨ v  
 fze ← (¯1 + d), t, k, fzb ((⊢/ (-∘≢ ⊢) ↑ ⊣), r, s, g, fzv, y, e ,∘⍪ l) ⊢  
 fzs ← (, 1 ↑ ⊢) (1 ⊖ (⊣ ((1 'Y', (2 ⌷ ⊣), ⊢) ⍪∘⍉∘⍪ (3 ↑ ⊣), ⊢) 1 ⌽ fzh, ¯1 ↓ 6 ↓ ⊣) ⍪ fze) (⌿∘⊢)  
 fz ← ((⊃⍪/) (1 ↑ ⊢), (((2 = d) (fzs ⍪ (1 ↓∘~ ⊣) (⌿∘⊢) 1 ↓ ⊢) ⊢)¨ 1 ↓ ⊢)) (1, 1 ↓ Sm) ⊂[0] ⊢  
 fd ← (1 ↑ ⊢) ⍪ ((1, 'Fd', 3 ↓ ⊢)⍤1 Fs) ⍪ 1 ↓ ⊢

tta ← (fc∘da∘(pc⍣≡)∘mr⍣≡)∘in⍣3∘sd∘lj∘ca∘fe∘lg∘nm∘rt∘mr∘dn∘lf∘du∘df∘rd∘rn  
 tt ← fd∘fz∘ff∘fg∘if∘ef∘vc∘rl∘av∘va∘lt∘nv∘fv∘ll∘ce∘ur∘tta

E1 ← {'fn' gcl ((⊂ n ,∘⊃ v), e, y) ⍵}  
 E2 ← {'fn' gcl ((⊂ n ,∘⊃ v), e, y) ⍵}  
 Ei ← {r l f ← ⊃ v ⍵ ⋄ ((⊃ n ⍵) ('fn' var) ⊃⊃ e ⍵), '=', ((⊃⊃ v ⍵) ('fn' var) 1 ⊃ ⊃ e ⍵), ';', nl}  
 O1 ← {'op' gcl ((⊂ n ,∘⊃ v), e, y) ⍵}  
 O2 ← {'op' gcl ((⊂ n ,∘⊃ v), e, y) ⍵}  
 O0 ← {''}  
 Of ← {'EF(', ('∆' ⎕R '\_\_' ⊃ n ⍵), ',', (⊃⊃ v ⍵), ');', nl}  
 Fd ← {'FP(', (⊃ n ⍵), ');', nl}  
 F0 ← {'DF(', (⊃ n ⍵), '\_f){', nl, 'A\*env[]={tenv};', nl}  
 F1 ← {'DF(', (⊃ n ⍵), '\_f){', nl, ('env0' dnv ⍵), (fnv ⍵)}  
 G0 ← { v ← (⊃⊃ v ⍵) ('' var) 1 ⊃ ⊃ e ⍵  
 'if(1!=cnt(', v, '))err(5);if(', v, '.v.as(s32).scalar<I>()){', nl}  
 G1 ← {'z=', ((⊃ n ⍵) ('' var) ⊃ ⊃ e ⍵), ';goto L', (⍕⊃ l ⍵), ';}', nl}  
 L0 ← {'z=', a, ';L', (⍕⊃ n ⍵), ':', (a ← (1 ⊃ ⊃ v ⍵) ('' var) 1 ⊃ ⊃ e ⍵), '=z;', nl}  
 Z0 ← {'}', nl, nl}  
 Z1 ← {'}', nl, nl}  
 Ze ← {'}', nl, nl}  
 M0 ← {rth, ('tenv' dnv ⍵), nl, 'A\*env[]={', ((0 ≡ ⊃⍵) ⊃ 'tenv' 'NULL'), '};', nl, nl}  
 S0 ← {(('{', rk0, srk, 'DO(i,prk)cnt\*=sp[i];', spp, sfv, slp) ⍵)}  
 Y0 ← {⊃,/ ((⍳ ≢ ⊃ n ⍵) ((⊣ sts¨ (⊃l) ,¨∘⊃ s), '}', nl, ⊣ ste¨ (⊃n) var¨∘⊃ r) ⍵), '}', nl}

gc ← {⊃,/ {0 = ⊃ t ⍵ : ⊂ 5 ⍴ ⍬ ⋄ ⊂ (⍎ (⊃ t ⍵), ⍕ ⊃ k ⍵) ⍵}⍤1 ⊢ ⍵}

syms ← ,¨ '+' '-' '×' '÷' '\*' '⍟' '|' '○' '⌊' '⌈' '!'  
 nams ← 'add' 'sub' 'mul' 'div' 'exp' 'log' 'res' 'cir' 'min' 'max' 'fac'  
 syms ,← ,¨ '<' '≤' '=' '≥' '>' '≠' '~' '∧' '∨' '⍲' '⍱'  
 nams ,← 'lth' 'lte' 'eql' 'gte' 'gth' 'neq' 'not' 'and' 'lor' 'nan' 'nor'  
 syms ,← ,¨ '⌷' '[' '⍳' '⍴' ',' '⍪' '⌽' '⍉' '⊖' '∊' '⊃'  
 nams ,← 'sqd' 'brk' 'iot' 'rho' 'cat' 'ctf' 'rot' 'trn' 'rtf' 'mem' 'dis'  
 syms ,← ,¨ '≡' '≢' '⊢' '⊣' '⊤' '⊥' '/' '⌿' '\' '⍀' '?'  
 nams ,← 'eqv' 'nqv' 'rgt' 'lft' 'enc' 'dec' 'red' 'rdf' 'scn' 'scf' 'rol'  
 syms ,← ,¨ '↑' '↓' '¨' '⍨' '.' '⍤' '⍣' '∘' '∪' '∩'  
 nams ,← 'tke' 'drp' 'map' 'com' 'dot' 'rnk' 'pow' 'jot' 'unq' 'int'  
 syms ,← ,¨ '⍋' '⍒' '∘.' '⍷' '⊂' '⌹' '⎕FFT' '⎕IFFT' '%u'   
 nams ,← 'gdu' 'gdd' 'oup' 'fnd' 'par' 'mdv' 'fft' 'ift' ''

nl ← ⎕UCS 13 10 ⋄ fvs ← ,⍤0 (⌿⍨) 0 ≠ (≢∘⍴¨ ⊣) ⋄ cln ← '¯' ⎕R '-' ⋄ cnm ← (syms ⍳ ⊂) ⊃ (nams, ⊂)  
 lits ← {'A(0,eshp,constant(', (cln ⍕ ⍵), ',eshp,', ('f64' 's32' ⊃⍨ ⍵ = ⌊⍵), '))'}  
 litv ← {'std::vector<', ('DI' ⊃⍨ ∧/ ⍵ = ⌊⍵), '>{', (cln ⊃ {⍺, ',', ⍵}/ ⍕¨ ⍵), '}.data()'}  
 lita ← {'A(1,dim4(', (⍕≢⍵), '),array(', (⍕≢⍵), ',', (litv ⍵), '))'}  
 lit ← {' ' = ⊃ 0 ⍴ ⍵ : (cnm ⍵), ⍺ ⋄ 1 = ≢⍵ : lits ⍵ ⋄ lita ⍵}  
 var ← {⍺ ≡ ,'⍺' : ,'l' ⋄ ⍺ ≡ ,'⍵' : ,'r' ⋄ ¯1 ≥ ⊃⍵ : ⍺⍺ lit ,⍺ ⋄ 'env[', (⍕⊃⍵), '][', (⍕⊃ ⌽⍵), ']'}  
 dnv ← {(0 ≡ z) ⊃ ('A ', ⍺, '[', (⍕ z ← ⊃ v ⍵), '];') ('A\*', ⍺, '=NULL;')}  
 fnv ← {z ← 'A\*env[', (⍕ 1 + ⊃ s ⍵), ']={', (⊃,/ (⊂'env0'), {',p[', (⍕⍵), ']'}¨ ⍳ ⊃ s ⍵), '};', nl}  
 gcl ← {z r l n ← ((3 ⍴ ⊂'fn'), ⊂⍺) {⊃ ⍺ var/ ⍵}¨ ↓ (⊃⍵), ⍪ 1 ⊃ ⍵ ⋄ n, '(', (⊃{⍺, ',', ⍵}/ z l r ~ ⊂'fn'), ',env);', nl}

∇ Z ← Gfx∆Init S  
 'w\_new' ⎕NA 'P ', (BSO S), '|w\_new <C[]'  
 'w\_close' ⎕NA 'I ', (BSO S),'|w\_close P'  
 'w\_del' ⎕NA (BSO S), '|w\_del P'  
 'w\_img' ⎕NA (BSO S), '|w\_img <PP P'  
 'w\_plot' ⎕NA (BSO S), '|w\_plot <PP P'  
 'w\_hist' ⎕NA (BSO S), '|w\_hist <PP F8 F8 P'  
 'loadimg' ⎕NA (BSO S), '|loadimg >PP <C[] I'  
 'saveimg' ⎕NA (BSO S), '|saveimg <PP <C[]'  
 Z ← 0 0 ⍴ ⍬  
∇

Display ← { ⍺ ← 'Co-dfns' ⋄ W ← w\_new ⊂⍺ ⋄ 777 :: w\_del W  
 w\_del W ⊣ W ⍺⍺{w\_close ⍺ : ⍎'⎕SIGNAL 777' ⋄ ⍺ ⍺⍺ ⍵}⍣⍵⍵ ⊢ ⍵}  
 LoadImage ← {⍺ ← 1 ⋄ ⍉ loadimg ⍬ ⍵ ⍺}  
 SaveImage ← {⍺ ← 'image.png' ⋄ saveimg (⍉⍵) ⍺}  
 Image ← {~ 2 3 ∨.= ≢⍴⍵ : ⎕SIGNAL 4 ⋄ (3 ≠ 2 ⊃ 3 ↑ ⍴⍵) ∧ 3 = ≢⍴⍵ : ⎕SIGNAL 5 ⋄ ⍵ ⊣ w\_img (⍉⍵) ⍺}  
 Plot ← {2 ≠ ≢⍴⍵ : ⎕SIGNAL 4 ⋄ ~ 2 3 ∨.= 1 ⊃ ⍴⍵ : ⎕SIGNAL 5 ⋄ ⍵ ⊣ w\_plot (⍉⍵) ⍺}  
 Histogram ← {⍵ ⊣ w\_hist ⍵, ⍺}

:EndNamespace