⍝ 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 ← 2018 1 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) ⍵}

f∆ ← 'ptknrsgvyeld' ⋄ N∆ ← 'ABEFGLMNOPVZ'  
⎕FX∘⍉∘⍪¨ f∆ ,¨ '←{'∘,¨ (⍕¨ ⍳≢f∆) ,¨ ⊂'⊃⍵}'  
⎕FX∘⍉∘⍪¨ N∆ ,¨ 'm←{'∘,¨ (⍕¨ ⍳≢N∆) ,¨ ⊂'=t⍵}'  
⎕FX∘⍉∘⍪¨ 'GLM' ,¨ '←{⍪/(0 '∘,¨ (⍕¨ N∆ ⍳ 'GLM') ,¨ ⊂' 0 0),1+@0⍉↑⍵}'  
⎕FX∘⍉∘⍪¨ 'ABEFO' ,¨ '←{⍪/(0 '∘,¨ (⍕¨ N∆ ⍳ 'ABEFO') ,¨ ⊂' ⍺⍺ 0),1+@0⍉↑⍵}'  
⎕FX∘⍉∘⍪¨ 'NPVZ' ,¨ '←{0(N∆⍳'''∘,¨ 'NPVZ' ,¨ ''')'∘,¨ '0(⍎⍵)' '0(⊂⍵)' '⍺⍺(⊂⍵)' '1(⊂⍵)' ,¨ '}'  
⎕FX∘⍉∘⍪¨ N∆ ,¨ ⊂'s←{⍵}' ⋄ at ← {⍺ ⍺⍺ ⍵⍵ ⍵} ⋄ new ← {⍵} ⋄ wrap ← {⍵}

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 ⍵, ⍺}

∇ r ← List  
 r ← ⎕NS¨ 1 ⍴ ⊂⍬ ⋄ r.Name ← ,¨ ⊂'Compile' ⋄ r.Group←⊂ 'CODFNS'   
 r[0].Desc ← 'Compile an object using Co-dfns'  
 r.Parse ← ⊂ '2S -af=cpu opencl cuda '   
∇

Convert ← {⍺ (⎕SE.SALT.Load '[SALT]/lib/NStoScript -noname').ntgennscode ⍵}  
Run ← { C I ← ⍵ ⋄ in out ← I.Arguments ⋄ AF∆LIB ∘← I.af '' ⊃⍨ I.af ≡ 0  
 S ← (⊂ ':Namespace ', out), 2 ↓ 0 0 0 out Convert ##.THIS.⍎ in  
 'Compile' ≡ C : {} {\_ ← {##.THIS.⍎ out, '←⍵'} out Fix S ⊣ ⎕EX '##.THIS.', out  
 ⎕CMD 'copy "%CUDA\_PATH%\nvvm\bin\nvvm64\*" /Y' /⍨ (I.af ≡ 'cuda') ∧ opsys 1 0 0} ⍬}  
Help ← {'Usage: <object> <target> [-af={cpu,opencl,cuda}]'}

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

\_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 \_s (0 Var \_some) \_as (3 A∘⌽)  
 Pex ← {⍺ (rpar \_s Ex \_s lpar) ⍵}  
 Atom ← Strand \_o (0 Var \_as (1 A)) \_o (Num \_some \_as (0 A∘⌽)) \_o Pex  
 Brk ← rbrk \_s {⍺ (Ex \_opt \_s (semi \_s (Ex \_opt) \_any)) ⍵} \_s lbrk \_as (3 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 + 3 3 2 2 ⊤ (6 4 4 ⌿ 1 5 9) + 2 × ⍳ 14  
 a ← ↓⍉↑ (e ,¨¨ ⊂⍺) Gex \_o Ex \_o Fex Stmts \_then Fn¨ ⊂⍵  
 m ← (0 = ⊃ a) ∧ ∧⌿ (∨⍀ ∘.=⍨ ⍳14) ∨ ∘.≢⍨ 1 ⊃ a  
 ~∨⌿ m : (⌈⌿ ⊃a) ⍬ ⍺ ⍵  
 (1 = +⌿ m) ∧ 2 > m ⍳ 1 : 0 (,⊂ 0 (N∆ ⍳ 'F') 1 0 ⍪¨ 1 +@0 ⊃⊃ m ⌿ 1 ⊃ a) ⍺ ⍵  
 z ← ⍪⌿↑ (⊂ 0 (N∆ ⍳ 'F') ¯1 0), ({1 (N∆ ⍳ 'F') ⍵ 0}¨ 1 + m ⌿ ⍳14) ⍪¨ (2 +@0 ⊃)¨ m ⌿ 1 ⊃ a   
 0 (,⊂z) ⍺ ⍵}  
 Fn ← { ns ← (n z) ⌿⍨ m ← (Fm ∧ ¯1 ∊⍨ k) ⊢ z ← ⍪⌿↑⍵ ⋄ 0 = ≢ns : 0 (,⊂z) ⍺ ''  
 r ← ↓⍉↑ ⍺∘Fa¨ ns ⋄ 0 < c ← ⌈⌿⊃r : c ⍬ ⍺ ⍵  
 z ← (⊂¨¨ z) ((⊃⍪⌿) ⊣@{m})¨⍨ ↓ (m ⌿ p z) +@0 ⍉↑ ⊃¨ 1 ⊃ r  
 0 (,⊂z) ⍺ ''}  
 Pfe ← {⍺ (rpar \_s Fex \_s lpar) ⍵}  
 Bfn ← rbrc Blrp lbrc \_as {0 (N∆ ⍳ 'F') ¯1 (,⊂⌽ 1 ↓ ¯1 ↓ ⍵)}  
 Fnp ← Prim \_o (1 Var) \_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 (⍪/ ⍳∘≢ +@0 ⍉∘↑∘⌽)) \_o Dop2 \_o Bop  
 Trn ← {⍺ (Afx \_s ((Afx \_o Idx \_o Atom) \_s (∇ \_opt) \_opt)) ⍵} \_as (3 F∘⌽)  
 Bind ← {⍺ (gets \_s (name \_as ⌽) \_env (⊣ ⍪¨⍨ ⍺⍺ ,⍨∘⊂ ⊢) \_as (0 (N∆ ⍳ 'B') ⍺⍺ ,∘⊂ ⊢)) ⍵}  
 Asgn ← gets \_s Brk \_s (name \_as ⌽ \_t (0 = Vt) \_as (0 V∘,∘⊃)) \_as (4 E∘⌽)  
 Fex ← Afx \_s (Trn \_opt) \_s (1 Bind \_any) \_as (⍪/ ⍳∘≢ +@0 ⍉∘↑∘⌽)  
 App ← Afx \_s (Idx \_o Atom \_s (dop2 \_not) \_opt) \_as {(≢⍵) E ⌽⍵}  
 Ex ← Idx \_o Atom \_s {⍺ (0 Bind \_o Asgn \_o App \_s ∇ \_opt) ⍵} \_as (⍪/ ⍳∘≢ +@0 ⍉∘↑∘⌽)  
 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 ← ⍬ ⍬ Ns ∊{⍵ /⍨ ∧\ '⍝' ≠ ⍵}¨⍵ ,¨ ⎕UCS 10 : ⎕SIGNAL c  
 (↓ s (-⍳)@3 ↑⊃a) e (s ← 0 (,'⍵') (,'⍺') '⍺⍺' '⍵⍵' (⊣, n ~ ⊣) ⊃a)}

⍝ A B E F G L M N O P V Z  
⍝ 0 1 2 3 4 5 6 7 8 9 10 11  
tt ← {d t k n ← ⍵ ⋄ I ← {(⊂⍵) ⌷ ⍺}  
 \_ ← 2 {0 ⊣ l[⍵[i]] ← ⍵[¯1 + i ← ⍸ 0, 2 =⌿ i] ⊣ p[⍵] ← ⍺[i ← ⍺ ⍸ ⍵]} ⌿ ⊢∘⊂⌸ d ⊣ p ← l ← ⍳≢ d  
 ⍝ Drop unnamed top-level functions  
 ⍝ Box mutated names  
 ⍝ Resolve names  
 gf ← {p I@{3 ≠ t[⍵]}⍣≡ p[⍵]}  
 bi ← ⍸ t = 1 ⋄ bv ← {⍵[bi ⍳ ⍵]}@{1 = t[⍵]}⍣≡ {⍵[⍋p[⍵]]} ⍸ 1 = t[p]  
 xi ← ⍸ 3 ≠ t[bf ← gf bi] ⋄ xv ← bv[xi] ⋄ xn ← n[bi[xi]]  
 gn ← { i f ← ⍵ ⋄ v ← ⍸,⊃ n[i] f ∧.(∘.=) n[bi] bf ⋄ n[i[a ← ⌊ v ÷ ≢bv]] ← bv[(≢bv) | v] ⋄ i (gf f) I¨ ⊂ (⍳≢i) ~ a}  
 \_ ← gn⍣{0 = ≢⊃⍺} {⍵ (gf ⍵)} ⍸ (n < ¯4) ∧ t = 10  
 n ← bi (⊢ - 1 + ⍸) n[nb ← ⍸ t ≠ 1]  
 p ← bi (⊢ - 1 + ⍸) p I@{1 = t[⍵]}⍣≡ p[nb]  
 l ← bi (⊢ - 1 + ⍸) nb I⍨ {bv[bi ⍳ ⍵]}@{1 = t[⍵]} l[bi[i]]@(bv[i]) ⊢ l ⊣ i ← ⍸ l[bi] ≠ bi  
 t ← t[nb] ⋄ k ← k[nb] ⋄ xv ← xv - 1 + bi ⍸ xv  
 ⍝ Lift guard test expressions  
 ⍝ Flatten expressions  
 ⍝ Label jumps  
 ⍝ Lift functions  
 ⍝ Inline functions  
 ⍝ Propagate constants  
 ⍝ Fold constants  
 ⍝ Dead, useless code elimination  
 ⍝ Allocate frames  
 ⍝ Create Init function  
 ⍝ Declare functions  
 ⍝ Sort AST  
 ⍝ Flatten AST

p t k n l xn xv  
}

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}

rth ← {⊃,/ (⊂nl) ,¨⍨ 2 ↓¨ ¯2 ↓ c ↓⍨ 1 + (⊂'rth') ⍳⍨ 3 ↑¨ c ← ⎕SRC ⎕THIS}

⍝ #include <time.h>

⍝ #include <stdint.h>

⍝ #include <inttypes.h>

⍝ #include <limits.h>

⍝ #include <float.h>

⍝ #include <math.h>

⍝ #include <memory>

⍝ #include <algorithm>

⍝ #include <string>

⍝ #include <cstring>

⍝ #include <vector>

⍝ #include <unordered\_map>

⍝ #include <arrayfire.h>

⍝ using namespace af;

⍝

⍝ #if AF\_API\_VERSION < 35

⍝ #error "Your ArrayFire version is too old."

⍝ #endif

⍝ #ifdef \_WIN32

⍝ #define EXPORT extern "C" \_\_declspec(dllexport)

⍝ #elif defined(\_\_GNUC\_\_)

⍝ #define EXPORT extern "C" \_\_attribute\_\_ ((visibility ("default")))

⍝ #else

⍝ #define EXPORT extern "C"

⍝ #endif

⍝ #ifdef \_MSC\_VER

⍝ #define RSTCT \_\_restrict

⍝ #else

⍝ #define RSTCT restrict

⍝ #endif

⍝ #define S struct

⍝ #define Z static

⍝ #define R return

⍝ #define RANK(lp) ((lp)->p->r)

⍝ #define TYPE(lp) ((lp)->p->t)

⍝ #define SHAPE(lp) ((lp)->p->s)

⍝ #define ETYPE(lp) ((lp)->p->e)

⍝ #define DATA(lp) ((V\*)&SHAPE(lp)[RANK(lp)])

⍝ #define CS(n,x) case n:x;break;

⍝ #define DO(n,x) {I i=0,\_i=(n);for(;i<\_i;++i){x;}}

⍝ #define DOB(n,x) {B i=0,\_i=(n);for(;i<\_i;++i){x;}}

⍝

⍝ typedef enum{APLNC=0,APLU8,APLTI,APLSI,APLI,APLD,APLP,APLU,APLV,APLW,APLZ,

⍝ APLR,APLF,APLQ}APLTYPE;

⍝ typedef long long L;typedef int I;typedef int16\_t S16;typedef int8\_t S8;

⍝ typedef double D;typedef unsigned char U8;typedef unsigned U;

⍝ typedef dim\_t B;typedef cdouble DZ;typedef void V;typedef std::string STR;

⍝

⍝ S{U f=3;U n;U x=0;wchar\_t\*v=L"Co-dfns";const wchar\_t\*e;V\*c;}dmx;

⍝ S lp{S{L l;B c;U t:4;U r:4;U e:4;U \_:13;U \_1:16;U \_2:16;B s[1];}\*p;};

⍝ S dwa{B z;S{B z;V\*(\*ga)(U,U,B\*,S lp\*);V(\*p[16])();V(\*er)(V\*);}\*ws;V\*p[4];};

⍝ S dwa\*dwafns;Z V derr(U n){dmx.n=n;dwafns->ws->er(&dmx);}

⍝ EXPORT I DyalogGetInterpreterFunctions(dwa\*p){

⍝ if(p)dwafns=p;else R 0;if(dwafns->z<sizeof(S dwa))R 16;R 0;}

⍝ Z V err(U n,wchar\_t\*e){dmx.e=e;throw n;}Z V err(U n){dmx.e=L"";throw n;}

⍝ S A{I r;dim4 s;array v;A(I r,dim4 s,array v):r(r),s(s),v(v){}

⍝ A():r(0),s(dim4()),v(array()){}};

⍝ int isinit=0;dim4 eshp=dim4(0,(B\*)NULL);std::wstring msg;

⍝ #define NM(n,nm,sm,sd,di,mf,df,ma,da) S n##\_f:FN{di;mf;df;ma;da;\

⍝ n##\_f(STR s,I m,I d):FN(s,m,d){}} n##fn(nm,sm,sd);

⍝ #define OM(n,nm,sm,sd,mf,df) S n##\_o:MOP{mf;df;\

⍝ n##\_o(FN&l,A\*p[]):MOP(nm,sm,sd,l,p){}};

⍝ #define OD(n,nm,sm,sd,mf,df) S n##\_o:DOP{mf;df;\

⍝ n##\_o(FN&l,FN&r,A\*p[]):DOP(nm,sm,sd,l,r,p){}\

⍝ n##\_o(const A&l,FN&r,A\*p[]):DOP(nm,sm,sd,l,r,p){}\

⍝ n##\_o(FN&l,const A&r,A\*p[]):DOP(nm,sm,sd,l,r,p){}};

⍝ #define MT

⍝ #define DID inline array id(dim4)

⍝ #define MFD inline V operator()(A&,const A&,A\*[])

⍝ #define MAD inline V operator()(A&,const A&,D,A\*[])

⍝ #define DFD inline V operator()(A&,const A&,const A&,A\*[])

⍝ #define DAD inline V operator()(A&,const A&,const A&,D,A\*[])

⍝ #define DI(n) inline array n::id(dim4 s)

⍝ #define ID(n,x,t) DI(n##\_f){R constant(x,s,t);}

⍝ #define MF(n) inline V n::operator()(A&z,const A&r,A\*p[])

⍝ #define MA(n) inline V n::operator()(A&z,const A&r,D ax,A\*p[])

⍝ #define DF(n) inline V n::operator()(A&z,const A&l,const A&r,A\*p[])

⍝ #define DA(n) inline V n::operator()(A&z,const A&l,const A&r,D ax,A\*p[])

⍝ #define SF(n,x) inline V n::operator()(A&z,const A&l,const A&r,A\*p[]){\

⍝ if(l.r==r.r&&l.s==r.s){\

⍝ z.r=l.r;z.s=l.s;const array&lv=l.v;const array&rv=r.v;x;R;}\

⍝ if(!l.r){\

⍝ z.r=r.r;z.s=r.s;const array&rv=r.v;array lv=tile(l.v,r.s);x;R;}\

⍝ if(!r.r){\

⍝ z.r=l.r;z.s=l.s;array rv=tile(r.v,l.s);const array&lv=l.v;x;R;}\

⍝ if(l.r!=r.r)err(4);if(l.s!=r.s)err(5);err(99);}

⍝ #define FP(n) NM(n,"",0,0,MT,MFD,DFD,MT,MT);MF(n##\_f){n##fn(z,A(),r,p);}

⍝ #define EF(n,m) EXPORT V n##\_dwa(lp\*z,lp\*l,lp\*r){try{\

⍝ A cl,cr,za;if(!isinit){Initfn(za,cl,cr,NULL);isinit=1;}\

⍝ cpda(cr,r);if(l!=NULL)cpda(cl,l);m##fn(za,cl,cr,env);cpad(z,za);}\

⍝ catch(U n){derr(n);}\

⍝ catch(exception e){msg=mkstr(e.what());dmx.e=msg.c\_str();derr(500);}}\

⍝ EXPORT V n##\_cdf(A\*z,A\*l,A\*r){try{m##fn(\*z,\*l,\*r,env);}catch(U n){derr(n);}\

⍝ catch(exception x){msg=mkstr(x.what());dmx.e=msg.c\_str();derr(500);}}

⍝

⍝ S FN{STR nm;I sm;I sd;FN(STR nm,I sm,I sd):nm(nm),sm(sm),sd(sd){}

⍝ FN():nm(""),sm(0),sd(0){}

⍝ virtual array id(dim4 s){err(16);R array();}

⍝ virtual V operator()(A&z,const A&r,A\*p[]){err(99);}

⍝ virtual V operator()(A&z,const A&r,D ax,A\*p[]){err(99);}

⍝ virtual V operator()(A&z,const A&l,const A&r,A\*p[]){err(99);}

⍝ virtual V operator()(A&z,const A&l,const A&r,D ax,A\*p[]){err(99);}};

⍝ FN MTFN;

⍝ S MOP:FN{FN&ll;A\*\*pp;

⍝ MOP(STR nm,I sm,I sd,FN&ll,A\*pp[]):FN(nm,sm,sd),ll(ll),pp(pp){}};

⍝ S DOP:FN{I fl;I fr;FN&ll;A aa;FN&rr;A ww;A\*\*pp;

⍝ DOP(STR nm,I sm,I sd,FN&l,FN&r,A\*p[])

⍝ :FN(nm,sm,sd),fl(1),fr(1),ll(l),aa(A()),rr(r),ww(A()),pp(p){}

⍝ DOP(STR nm,I sm,I sd,A l,FN&r,A\*p[])

⍝ :FN(nm,sm,sd),fl(0),fr(1),ll(MTFN),aa(l),rr(r),ww(A()),pp(p){}

⍝ DOP(STR nm,I sm,I sd,FN&l,A r,A\*p[])

⍝ :FN(nm,sm,sd),fl(1),fr(0),ll(l),aa(A()),rr(MTFN),ww(r),pp(p){}};

⍝

⍝ std::wstring mkstr(const char\*s){B c=std::strlen(s);std::wstring t(c,L' ');

⍝ mbstowcs(&t[0],s,c);R t;}

⍝ I scm(FN&f){R f.sm;}I scm(const A&a){R 1;}

⍝ I scd(FN&f){R f.sd;}I scd(const A&a){R 1;}

⍝ B cnt(dim4 s){B c=1;DO(4,c\*=s[i]);R c;}

⍝ B cnt(const A&a){B c=1;DO(a.r,c\*=a.s[i]);R c;}

⍝ B cnt(lp\*d){B c=1;DO(RANK(d),c\*=SHAPE(d)[i]);R c;}

⍝ array scl(I x){R constant(x,dim4(1),s32);}

⍝ A scl(array v){R A(0,dim4(1),v);}

⍝ dtype mxt(dtype at,dtype bt){if(at==c64||bt==c64)R c64;

⍝ if(at==f64||bt==f64)R f64;

⍝ if(at==s32||bt==s32)R s32;if(at==s16||bt==s16)R s16;

⍝ if(at==b8||bt==b8)R b8;err(16);R f64;}

⍝ dtype mxt(const array&a,const array&b){R mxt(a.type(),b.type());}

⍝ dtype mxt(dtype at,const A&b){R mxt(at,b.v.type());}

⍝ Z array da16(B c,dim4 s,lp\*d){std::vector<S16>b(c);

⍝ S8\*v=(S8\*)DATA(d);DOB(c,b[i]=v[i]);R array(s,b.data());}

⍝ Z array da8(B c,dim4 s,lp\*d){std::vector<char>b(c);

⍝ U8\*v=(U8\*)DATA(d);DOB(c,b[i]=1&(v[i/8]>>(7-(i%8))))

⍝ R array(s,b.data());}

⍝ V cpad(lp\*d,A&a){I t;B c=cnt(a);

⍝ switch(a.v.type()){CS(c64,t=APLZ);

⍝ CS(s32,t=APLI);CS(s16,t=APLSI);CS(b8,t=APLTI);CS(f64,t=APLD);

⍝ default:if(c)err(16);t=APLI;}

⍝ B s[4];DO(a.r,s[a.r-(i+1)]=a.s[i]);dwafns->ws->ga(t,a.r,s,d);

⍝ if(c)a.v.host(DATA(d));}

⍝ V cpda(A&a,lp\*d){if(15!=TYPE(d))err(16);if(4<RANK(d))err(16);

⍝ dim4 s(1);DO(RANK(d),s[RANK(d)-(i+1)]=SHAPE(d)[i]);B c=cnt(d);

⍝ switch(ETYPE(d)){

⍝ CS(APLZ,a=A(RANK(d),s,c?array(s,(DZ\*)DATA(d)):scl(0)))

⍝ CS(APLI,a=A(RANK(d),s,c?array(s,(I\*)DATA(d)):scl(0)))

⍝ CS(APLD,a=A(RANK(d),s,c?array(s,(D\*)DATA(d)):scl(0)))

⍝ CS(APLSI,a=A(RANK(d),s,c?array(s,(S16\*)DATA(d)):scl(0)))

⍝ CS(APLTI,a=A(RANK(d),s,c?da16(c,s,d):scl(0)))

⍝ CS(APLU8,a=A(RANK(d),s,c?da8(c,s,d):scl(0)))

⍝ default:err(16);}}

⍝

⍝ NM(add,"add",1,1,DID,MFD,DFD,MT ,MT )NM(sub,"sub",1,1,DID,MFD,DFD,MT ,MT )

⍝ NM(mul,"mul",1,1,DID,MFD,DFD,MT ,MT )NM(div,"div",1,1,DID,MFD,DFD,MT ,MT )

⍝ NM(max,"max",1,1,DID,MFD,DFD,MT ,MT )NM(min,"min",1,1,DID,MFD,DFD,MT ,MT )

⍝ NM(exp,"exp",1,1,DID,MFD,DFD,MT ,MT )NM(log,"log",1,1,MT ,MFD,DFD,MT ,MT )

⍝ NM(fac,"fac",1,1,DID,MFD,DFD,MT ,MT )NM(res,"res",1,1,DID,MFD,DFD,MT ,MT )

⍝ NM(and,"and",1,1,DID,MT ,DFD,MT ,MT )NM(lor,"lor",1,1,DID,MT ,DFD,MT ,MT )

⍝ NM(lth,"lth",1,1,DID,MT ,DFD,MT ,MT )NM(lte,"lte",1,1,DID,MT ,DFD,MT ,MT )

⍝ NM(gth,"gth",1,1,DID,MT ,DFD,MT ,MT )NM(gte,"gte",1,1,DID,MT ,DFD,MT ,MT )

⍝ NM(eql,"eql",1,1,DID,MT ,DFD,MT ,MT )NM(neq,"neq",1,1,DID,MT ,DFD,MT ,MT )

⍝ NM(nan,"nan",1,1,MT ,MT ,DFD,MT ,MT )NM(nor,"nor",1,1,MT ,MT ,DFD,MT ,MT )

⍝ NM(cir,"cir",1,1,MT ,MFD,DFD,MT ,MT )NM(not,"not",1,0,MT ,MFD,DFD,MT ,MT )

⍝ NM(rot,"rot",0,0,DID,MFD,DFD,MT ,MT )NM(rtf,"rtf",0,0,DID,MFD,DFD,MT ,MT )

⍝ NM(red,"red",0,0,DID,MT ,DFD,MT ,MT )NM(rdf,"rdf",0,0,DID,MT ,DFD,MT ,MT )

⍝ NM(scn,"scn",0,0,DID,MT ,DFD,MT ,MT )NM(scf,"scf",0,0,DID,MT ,DFD,MT ,MT )

⍝ NM(enc,"enc",0,0,DID,MT ,DFD,MT ,MT )NM(dec,"dec",0,0,MT ,MT ,DFD,MT ,MT )

⍝ NM(sqd,"sqd",0,0,MT ,MFD,DFD,MT ,MT )NM(brk,"brk",0,0,MT ,MFD,DFD,MT ,MT )

⍝ NM(iot,"iot",0,0,MT ,MFD,DFD,MT ,MT )NM(rho,"rho",0,0,MT ,MFD,DFD,MT ,MT )

⍝ NM(cat,"cat",0,0,MT ,MFD,DFD,MT ,DAD)NM(ctf,"ctf",0,0,MT ,MFD,DFD,MT ,MT )

⍝ NM(trn,"trn",0,0,MT ,MFD,DFD,MT ,MT )NM(rol,"rol",1,0,MT ,MFD,DFD,MT ,MT )

⍝ NM(dis,"dis",0,0,MT ,MFD,DFD,MT ,MT )NM(par,"par",0,0,MT ,MFD,DFD,MT ,MT )

⍝ NM(eqv,"eqv",0,0,MT ,MFD,DFD,MT ,MT )NM(nqv,"nqv",0,0,MT ,MFD,DFD,MT ,MT )

⍝ NM(rgt,"rgt",0,0,MT ,MFD,DFD,MT ,MT )NM(lft,"lft",0,0,MT ,MFD,DFD,MT ,MT )

⍝ NM(tke,"tke",0,0,MT ,MFD,DFD,MT ,MT )NM(drp,"drp",0,0,MT ,MFD,DFD,MT ,MT )

⍝ NM(unq,"unq",0,0,MT ,MFD,DFD,MT ,MT )NM(int,"int",0,0,MT ,MT ,DFD,MT ,MT )

⍝ NM(gdu,"gdu",0,0,MT ,MFD,DFD,MT ,MT )NM(gdd,"gdd",0,0,MT ,MFD,DFD,MT ,MT )

⍝ NM(mem,"mem",0,0,MT ,MFD,DFD,MT ,MT )NM(fnd,"fnd",0,0,MT ,MT ,DFD,MT ,MT )

⍝ NM(fft,"fft",1,0,MT ,MFD,MT ,MT ,MT )NM(ift,"ift",1,0,MT ,MFD,MT ,MT ,MT )

⍝ NM(mdv,"mdv",1,0,MT ,MFD,DFD,MT ,MT )

⍝

⍝ ID(add,0,s32)ID(sub,0,s32)ID(mul,1,s32)ID(div,1,s32)ID(res,0,s32)

⍝ ID(min,DBL\_MAX,f64)ID(max,-DBL\_MAX,f64)ID(exp,1,s32)ID(fac,1,s32)

⍝ ID(and,1,s32)ID(lor,0,s32)ID(lth,0,s32)ID(lte,1,s32)ID(eql,1,s32)

⍝ ID(gth,0,s32)ID(gte,1,s32)ID(neq,0,s32)ID(enc,0,s32)ID(red,1,s32)

⍝ ID(rdf,1,s32)ID(scn,1,s32)ID(scf,1,s32)ID(rot,0,s32)ID(rtf,0,s32)

⍝

⍝ OD(brk,"brk",scm(l),scd(l),MFD,DFD)

⍝ OM(com,"com",scm(l),scd(l),MFD,DFD)

⍝ OD(dot,"dot",0,0,MT,DFD)

⍝ OD(jot,"jot",(scm(l)&&scm(r)),(scd(l)&&scd(r)),MFD,DFD)

⍝ OM(map,"map",1,1,MFD,DFD)

⍝ OM(oup,"oup",0,0,MT,DFD)

⍝ OD(pow,"pow",scm(l),scd(l),MFD,DFD)

⍝ OM(red,"red",0,0,MFD,DFD)

⍝ OM(rdf,"rdf",0,0,MFD,DFD)

⍝ OD(rnk,"rnk",scm(l),0,MFD,DFD)

⍝ OM(scn,"scn",1,1,MFD,MT)

⍝ OM(scf,"scf",1,1,MFD,MT)

⍝

⍝ MF(add\_f){z=r;}

⍝ SF(add\_f,z.v=lv+rv)

⍝ SF(and\_f,if(lv.isbool()&&rv.isbool())z.v=lv&&rv;

⍝ else if(allTrue<I>(lv>=0&&lv<=1&&rv>0&&rv<=1))z.v=lv&&rv;

⍝ else{A a(z.r,z.s,lv);A b(z.r,z.s,rv);

⍝ lorfn(a,a,b,p);z.v=lv\*(rv/((!a.v)+a.v));})

⍝ MF(brk\_f){err(16);}

⍝ DF(brk\_f){if(l.r!=1)err(16);

⍝ z.r=r.r;z.s=r.s;z.v=l.v(r.v.as(s32));}

⍝ MF(cat\_f){z.r=1;z.s[0]=cnt(r);z.v=flat(r.v);}

⍝ DA(cat\_f){A nl=l,nr=r;I fx=(I)ceil(ax);

⍝ if(fx<0||(fx>r.r&&fx>l.r))err(4);

⍝ if(ax!=fx){if(r.r>3||l.r>3)err(10);

⍝ if(nl.r){nl.r++;DO(3-fx,nl.s[3-i]=nl.s[3-(i+1)]);nl.s[fx]=1;}

⍝ if(nr.r){nr.r++;DO(3-fx,nr.s[3-i]=nr.s[3-(i+1)]);nr.s[fx]=1;}

⍝ if(nl.r)nl.v=moddims(nl.v,nl.s);if(nr.r)nr.v=moddims(nr.v,nr.s);

⍝ catfn(z,nl,nr,fx,p);R;}

⍝ if(fx>=r.r&&fx>=l.r)err(4);

⍝ if(l.r&&r.r&&std::abs((I)l.r-(I)r.r)>1)err(4);

⍝ z.r=(l.r>=r.r)\*l.r+(r.r>l.r)\*r.r+(!r.r&&!l.r);

⍝ dim4 ls=l.s;dim4 rs=r.s;

⍝ if(!l.r){ls=rs;ls[fx]=1;}if(!r.r){rs=ls;rs[fx]=1;}

⍝ if(r.r&&l.r>r.r){DO(3-fx,rs[3-i]=rs[3-(i+1)]);rs[fx]=1;}

⍝ if(l.r&&r.r>l.r){DO(3-fx,ls[3-i]=ls[3-(i+1)]);ls[fx]=1;}

⍝ DO(4,if(i!=fx&&rs[i]!=ls[i])err(5));

⍝ DO(4,z.s[i]=(l.r>=r.r||i==fx)\*ls[i]+(r.r>l.r||i==fx)\*rs[i]);

⍝ if(!cnt(l)){z.v=r.v;R;}if(!cnt(r)){z.v=l.v;R;}

⍝ dtype mt=mxt(r.v,l.v);

⍝ array lv=(l.r?moddims(l.v,ls):tile(l.v,ls)).as(mt);

⍝ array rv=(r.r?moddims(r.v,rs):tile(r.v,rs)).as(mt);

⍝ z.v=join(fx,lv,rv);}

⍝ DF(cat\_f){if(l.r||r.r){catfn(z,l,r,0,p);R;}

⍝ A a,b;catfn(a,l,p);catfn(b,r,p);catfn(z,a,b,0,p);}

⍝ MF(cir\_f){z.r=r.r;z.s=r.s;z.v=Pi\*r.v.as(f64);}

⍝ SF(cir\_f,array fv=rv.as(f64);

⍝ if(!l.r){I x=l.v.as(s32).scalar<I>();if(abs(x)>10)err(16);

⍝ switch(x){CS(0,z.v=sqrt(1-fv\*fv))CS(1,z.v=sin(fv))CS(2,z.v=cos(fv))

⍝ CS(3,z.v=tan(fv))CS(4,z.v=sqrt(1+fv\*fv))CS(5,z.v=sinh(fv))

⍝ CS(6,z.v=cosh(fv))CS(7,z.v=tanh(fv))CS(8,z.v=sqrt(fv\*fv-1))CS(9,z.v=fv)

⍝ CS(10,z.v=abs(fv))CS(-1,z.v=asin(fv))CS(-2,z.v=acos(fv))

⍝ CS(-3,z.v=atan(fv))CS(-4,z.v=(fv+1)\*sqrt((fv-1)/(fv+1)))

⍝ CS(-5,z.v=asinh(fv))CS(-6,z.v=acosh(fv))CS(-7,z.v=atanh(fv))

⍝ CS(-8,z.v=-sqrt(fv\*fv-1))CS(-9,z.v=fv)CS(-10,z.v=fv)}R;}

⍝ if(anyTrue<I>(abs(lv)>10))err(16);B c=cnt(z);std::vector<I> a(c);

⍝ std::vector<D> b(c);lv.as(s32).host(a.data());fv.host(b.data());

⍝ std::vector<D> zv(c);

⍝ DOB(c,switch(a[i]){CS(0,zv[i]=sqrt(1-b[i]\*b[i]))CS(1,zv[i]=sin(b[i]))

⍝ CS(2,zv[i]=cos(b[i]))CS(3,zv[i]=tan(b[i]))CS(4,zv[i]=sqrt(1+b[i]\*b[i]))

⍝ CS(5,zv[i]=sinh(b[i]))CS(6,zv[i]=cosh(b[i]))CS(7,zv[i]=tanh(b[i]))

⍝ CS(8,zv[i]=sqrt(b[i]\*b[i]-1))CS(9,zv[i]=b[i])CS(10,zv[i]=abs(b[i]))

⍝ CS(-1,zv[i]=asin(b[i]))CS(-2,zv[i]=acos(b[i]))CS(-3,zv[i]=atan(b[i]))

⍝ CS(-4,zv[i]=(b[i]==-1)?0:(b[i]+1)\*sqrt((b[i]-1)/(b[i]+1)))

⍝ CS(-5,zv[i]=asinh(b[i]))CS(-6,zv[i]=acosh(b[i]))CS(-7,zv[i]=atanh(b[i]))

⍝ CS(-8,zv[i]=-sqrt(b[i]\*b[i]-1))CS(-9,zv[i]=b[i])CS(-10,zv[i]=b[i])})

⍝ z.v=array(z.s,zv.data());)

⍝ MF(ctf\_f){dim4 sp=z.s;sp[1]=r.r?r.s[r.r-1]:1;sp[0]=sp[1]?cnt(r)/sp[1]:1;

⍝ sp[2]=sp[3]=1;z.r=2;z.s=sp;z.v=!cnt(z)?scl(0):array(r.v,z.s);}

⍝ DF(ctf\_f){I x=l.r>r.r?l.r:r.r;if(l.r||r.r){catfn(z,l,r,x-1,p);R;}

⍝ A a,b;catfn(a,l,p);catfn(b,r,p);catfn(z,a,b,0,p);}

⍝ DF(dec\_f){I ra=r.r?r.r-1:0;I la=l.r?l.r-1:0;z.r=ra+la;z.s=dim4(1);

⍝ if(l.s[0]!=1&&l.s[0]!=r.s[ra]&&r.s[ra]!=1)err(5);

⍝ DO(ra,z.s[i]=r.s[i])DO(la,z.s[i+ra]=l.s[i+1])

⍝ if(!cnt(z)){z.v=scl(0);R;}

⍝ if(!cnt(r)||!cnt(l)){z.v=constant(0,z.s,s32);R;}

⍝ B lc=l.s[0];array x=l.v;if(lc==1){lc=r.s[ra];x=tile(x,(I)lc);}

⍝ x=flip(scan(x,0,AF\_BINARY\_MUL,false),0);

⍝ x=array(x,lc,x.elements()/lc).as(f64);

⍝ array y=array(r.v,cnt(r)/r.s[ra],r.s[ra]).as(f64);

⍝ z.v=array(matmul(r.s[ra]==1?tile(y,1,(I)l.s[0]):y,x),z.s);}

⍝ MF(dis\_f){z.r=0;z.s=eshp;z.v=r.v(0);}

⍝ DF(dis\_f){if(l.v.isfloating())err(1);if(l.r>1)err(4);

⍝ B lc=cnt(l);if(!lc){z=r;R;}if(lc!=1||r.r!=1)err(4);

⍝ if(allTrue<char>(cnt(r)<=l.v(0)))err(3);

⍝ z.r=0;z.s=eshp;array i=l.v(0);z.v=r.v(i);}

⍝ MF(div\_f){z.r=r.r;z.s=r.s;z.v=1.0/r.v.as(f64);}

⍝ SF(div\_f,z.v=lv.as(f64)/rv.as(f64))

⍝ MF(drp\_f){if(r.r)err(16);z=r;}

⍝ DF(drp\_f){I lv[4];seq it[4];seq ix[4];B c=cnt(l);

⍝ if(l.r>1||(c>r.r&&r.r))err(4);if(!c){z=r;R;}

⍝ U rk=r.r?r.r:(U)l.s[0];z.r=rk;z.s=r.s;l.v.as(s32).host(lv);

⍝ DO((I)c,{U j=rk-(i+1);I a=std::abs(lv[i]);

⍝ if(a>=r.s[j]){z.s[j]=0;ix[j]=seq(0);it[j]=seq(0);}

⍝ else if(lv[i]<0){

⍝ z.s[j]=r.s[j]-a;ix[j]=seq((D)z.s[j]);it[j]=ix[j];}

⍝ else{z.s[j]=r.s[j]-a;ix[j]=seq(a,(D)r.s[j]-1);it[j]=ix[j]-(D)a;}})

⍝ if(!cnt(z)){z.v=scl(0);R;}z.v=array(z.s,r.v.type());z.v=0;

⍝ z.v(it[0],it[1],it[2],it[3])=r.v(ix[0],ix[1],ix[2],ix[3]);}

⍝ DF(enc\_f){I rk=r.r+l.r;if(rk>4)err(16);dim4 sp=r.s;DO(l.r,sp[i+r.r]=l.s[i])

⍝ if(!cnt(sp)){z.r=rk;z.s=sp;z.v=scl(0);R;}dim4 lt=sp,rt=sp;I k=l.r?l.r-1:0;

⍝ DO(r.r,rt[i]=1)DO(l.r,lt[i+r.r]=1)array rv=tile(r.v,rt);z.r=rk;z.s=sp;

⍝ array sv=flip(scan(flip(l.v,k),k,AF\_BINARY\_MUL),k);

⍝ array lv=tile(array(sv,rt),lt);af::index x[4];x[k]=0;

⍝ array dv=sv;dv(x[0],x[1],x[2],x[3])=1;I s[]={0,0,0,0};s[k]=-1;

⍝ dv=shift(dv,s[0],s[1],s[2],s[3]);dv=tile(array(dv,rt),lt);

⍝ z.v=(lv!=0)\*rem(rv,lv)+(lv==0)\*rv;z.v=(dv!=0)\*(z.v/dv).as(s32);}

⍝ SF(eql\_f,z.v=lv==rv)

⍝ MF(eqv\_f){z.r=0;z.s=eshp;z.v=scl(r.r!=0);}

⍝ DF(eqv\_f){z.r=0;z.s=eshp;

⍝ if(l.r==r.r&&l.s==r.s){z.v=allTrue(l.v==r.v);R;}z.v=scl(0);}

⍝ MF(exp\_f){z.r=r.r;z.s=r.s;z.v=exp(r.v.as(f64));}

⍝ SF(exp\_f,z.v=pow(lv.as(f64),rv.as(f64)))

⍝ MF(fac\_f){z.r=r.r;z.s=r.s;z.v=factorial(r.v.as(f64));}

⍝ SF(fac\_f,array lvf=lv.as(f64);array rvf=rv.as(f64);

⍝ z.v=exp(log(tgamma(lvf))+log(tgamma(rvf))-log(tgamma(lvf+rvf))))

⍝ MF(fft\_f){z.r=r.r;z.s=r.s;z.v=dft(r.v.type()==c64?r.v:r.v.as(c64),1,r.s);}

⍝ MF(ift\_f){z.r=r.r;z.s=r.s;z.v=idft(r.v.type()==c64?r.v:r.v.as(c64),1,r.s);}

⍝ DF(fnd\_f){A t(r.r,r.s,array(r.s,b8));if(!cnt(t)){t.v=scl(0);z=t;R;}

⍝ t.v=0;if(l.r>r.r){z=t;R;}DO(4,if(l.s[i]>r.s[i]){z=t;R;})

⍝ if(!cnt(l)){t.v=1;z=t;R;}dim4 sp;DO(4,sp[i]=1+(t.s[i]-l.s[i]))

⍝ seq x[4];DO(4,x[i]=seq((D)sp[i]))t.v(x[0],x[1],x[2],x[3])=1;

⍝ DO((I)l.s[0],I m=i;

⍝ DO((I)l.s[1],I k=i;

⍝ DO((I)l.s[2],I j=i;

⍝ DO((I)l.s[3],t.v(x[0],x[1],x[2],x[3])=t.v(x[0],x[1],x[2],x[3])

⍝ &(tile(l.v(m,k,j,i),sp)

⍝ ==r.v(x[0]+(D)m,x[1]+(D)k,x[2]+(D)j,x[3]+(D)i))))))

⍝ z=t;}

⍝

⍝ MF(gdd\_f){if(r.r<1)err(4);z.r=1;z.s=dim4(r.s[r.r-1]);

⍝ if(!cnt(r)){z.v=r.v;R;}I c=1;DO(r.r-1,c\*=(I)r.s[i]);

⍝ array mt,a=array(r.v,c,r.s[r.r-1]);z.v=iota(z.s,dim4(1),s32);

⍝ DO(c,sort(mt,z.v,flat(a(c-(i+1),z.v)),z.v,0,false))}

⍝ DF(gdd\_f){err(16);}

⍝ MF(gdu\_f){if(r.r<1)err(4);z.r=1;z.s=dim4(r.s[r.r-1]);

⍝ if(!cnt(r)){z.v=r.v;R;}I c=1;DO(r.r-1,c\*=(I)r.s[i]);

⍝ array mt,a=array(r.v,c,r.s[r.r-1]);z.v=iota(z.s,dim4(1),s32);

⍝ DO(c,sort(mt,z.v,flat(a(c-(i+1),z.v)),z.v,0,true))}

⍝ DF(gdu\_f){err(16);}

⍝ SF(gte\_f,z.v=lv>=rv)

⍝ SF(gth\_f,z.v=lv>rv)

⍝ DF(int\_f){if(r.r>1||l.r>1)err(4);

⍝ if(!cnt(r)||!cnt(l)){z.v=scl(0);z.s=dim4(0);z.r=1;R;}

⍝ dtype mt=mxt(l.v,r.v);z.v=setIntersect(l.v.as(mt),r.v.as(mt));

⍝ z.r=1;z.s=dim4(z.v.elements());}

⍝ MF(iot\_f){if(r.r>1)err(4);B c=cnt(r);if(c>4)err(10);

⍝ if(c>1)err(16);

⍝ z.r=1;z.s=dim4(r.v.as(s32).scalar<I>());

⍝ z.v=z.s[0]?iota(z.s,dim4(1),s32):scl(0);}

⍝ DF(iot\_f){z.r=r.r;z.s=r.s;B c=cnt(r);if(!c){z.v=scl(0);R;}

⍝ B lc=cnt(l)+1;if(lc==1){z.v=scl(0);R;};if(l.r>1)err(16);

⍝ array rf=flat(r.v).T();dtype mt=mxt(l.v,rf);

⍝ z.v=join(0,tile(l.v,1,(U)c).as(mt),rf.as(mt))==tile(rf,(U)lc,1);

⍝ z.v=min((z.v\*iota(dim4(lc),dim4(1,c),s32)+((!z.v)\*lc).as(s32)),0);

⍝ z.v=array(z.v,z.s);}

⍝ MF(lft\_f){z=r;}

⍝ DF(lft\_f){z=l;}

⍝ MF(log\_f){z.r=r.r;z.s=r.s;z.v=log(r.v.as(f64));}

⍝ SF(log\_f,z.v=log(rv.as(f64))/log(lv.as(f64)))

⍝ SF(lor\_f,if(rv.isbool()&&lv.isbool())z.v=lv||rv;

⍝ else if(lv.isbool()&&rv.isinteger())z.v=lv+(!lv)\*abs(rv).as(rv.type());

⍝ else if(rv.isbool()&&lv.isinteger())z.v=rv+(!rv)\*abs(lv).as(lv.type());

⍝ else if(lv.isinteger()&&rv.isinteger()){B c=cnt(z);

⍝ std::vector<I> a(c);abs(lv).as(s32).host(a.data());

⍝ std::vector<I> b(c);abs(rv).as(s32).host(b.data());

⍝ DOB(c,while(b[i]){I t=b[i];b[i]=a[i]%b[i];a[i]=t;})

⍝ z.v=array(z.s,a.data());}

⍝ else{B c=cnt(z);

⍝ std::vector<D> a(c);abs(lv).as(f64).host(a.data());

⍝ std::vector<D> b(c);abs(rv).as(f64).host(b.data());

⍝ DOB(c,while(b[i]>1e-12){D t=b[i];b[i]=fmod(a[i],b[i]);a[i]=t;})

⍝ z.v=array(z.s,a.data());})

⍝ SF(lte\_f,z.v=lv<=rv)

⍝ SF(lth\_f,z.v=lv<rv)

⍝ MF(max\_f){z.r=r.r;z.s=r.s;z.v=ceil(r.v).as(r.v.type());}

⍝ SF(max\_f,z.v=max(lv,rv))

⍝ MF(mem\_f){z.r=1;z.s=dim4(cnt(r));z.v=flat(r.v);}

⍝ DF(mem\_f){z.r=l.r;z.s=l.s;I lc=(I)cnt(z);if(!lc){z.v=scl(0);R;}

⍝ if(!cnt(r)){z.v=array(z.s,b8);z.v=0;R;}

⍝ array y=setUnique(flat(r.v));I rc=(I)y.elements();

⍝ array x=array(flat(l.v),lc,1);y=array(y,1,rc);

⍝ z.v=array(anyTrue(tile(x,1,rc)==tile(y,lc,1),1),z.s);}

⍝ MF(mdv\_f){if(r.r>2)err(4);if(r.r==2&&r.s[1]<r.s[0])err(5);if(!cnt(r))err(5);

⍝ if(r.s[0]==r.s[1]){z.r=r.r;z.s=r.s;z.v=inverse(r.v);R;}

⍝ if(r.r==1){z.v=matmulNT(inverse(matmulTN(r.v,r.v)),r.v);z.r=r.r;z.s=r.s;R;}

⍝ z.v=matmulTN(inverse(matmulNT(r.v,r.v)),r.v);z.r=r.r;z.s=r.s;

⍝ B k=z.s[0];z.s[0]=z.s[1];z.s[1]=k;z.v=transpose(z.v);}

⍝

⍝ DF(mdv\_f){if(r.r>2)err(4);if(l.r>2)err(4);if(r.r==2&&r.s[1]<r.s[0])err(5);

⍝ if(!cnt(r)||!cnt(l))err(5);if(r.r&&l.r&&l.s[l.r-1]!=r.s[r.r-1])err(5);

⍝ array rv=r.v,lv=l.v;if(r.r==1)rv=transpose(rv);if(l.r==1)lv=transpose(lv);

⍝ z.v=transpose(matmul(inverse(matmulNT(rv,rv)),matmulNT(rv,lv)));

⍝ z.r=(l.r-(l.r>0))+(r.r-(r.r>0));

⍝ if(l.r>1)z.s[0]=l.s[0];if(r.r>1)z.s[l.r>1]=r.s[0];}

⍝ MF(min\_f){z.r=r.r;z.s=r.s;z.v=floor(r.v).as(r.v.type());}

⍝ SF(min\_f,z.v=min(lv,rv))

⍝ MF(mul\_f){z.r=r.r;z.s=r.s;z.v=(r.v>0)-(r.v<0);}

⍝ SF(mul\_f,z.v=lv\*rv)

⍝ SF(nan\_f,z.v=!(lv&&rv))

⍝ SF(neq\_f,z.v=lv!=rv)

⍝ SF(nor\_f,z.v=!(lv||rv))

⍝ MF(not\_f){z.r=r.r;z.s=r.s;z.v=!r.v;}

⍝ DF(not\_f){err(16);}

⍝ MF(nqv\_f){z.v=scl(r.r?(I)r.s[r.r-1]:1);z.r=0;z.s=dim4(1);}

⍝ DF(nqv\_f){z.r=0;z.s=eshp;I t=l.r==r.r&&l.s==r.s;

⍝ if(t)t=allTrue<I>(l.v==r.v);z.v=scl(!t);}

⍝ MF(par\_f){err(16);}

⍝ DF(par\_f){err(16);}

⍝ DF(red\_f){if(l.r>1)err(4);z.r=r.r?r.r:1;z.s=r.s;

⍝ if(l.r!=0&&l.s[0]!=1&&r.r!=0&&r.s[0]!=1&&l.s[0]!=r.s[0])err(5);

⍝ array x=l.v;if(cnt(l)==1)x=tile(x,(I)r.s[0]);

⍝ array y=r.v;if(r.s[0]==1)y=tile(y,(I)cnt(l));

⍝ z.s[0]=sum<B>(abs(x));if(!cnt(z)){z.v=scl(0);R;}

⍝ array w=where(x).as(s32);

⍝ if(z.s[0]==w.elements()){z.v=y(w,span);R;}

⍝ array i=shift(accum(abs(x(w))),1),d=shift(w,1);i(0)=0;d(0)=0;

⍝ array v=array(z.s[0],s32),u=array(z.s[0],s32);v=0;u=0;

⍝ array s=(!sign(x(w))).as(s32);array t=shift(s,1);t(0)=0;

⍝ v(i)=w-d;u(i)=s-t;z.v=y(accum(v),span);

⍝ z.v\*=tile(accum(u),1,(I)z.s[1],(I)z.s[2],(I)z.s[3]);}

⍝ MF(res\_f){z.r=r.r;z.s=r.s;z.v=abs(r.v).as(r.v.type());}

⍝ SF(res\_f,z.v=rv-lv\*floor(rv.as(f64)/(lv+(0==lv))))

⍝ DF(rdf\_f){if(l.r>1)err(4);I ra=r.r?r.r-1:0;z.r=ra+1;z.s=r.s;

⍝ if(l.r!=0&&l.s[0]!=1&&r.r!=0&&r.s[ra]!=1&&l.s[0]!=r.s[ra])err(5);

⍝ array x=l.v;array y=r.v;if(cnt(l)==1)x=tile(x,(I)r.s[ra]);

⍝ if(r.s[ra]==1){dim4 s(1);s[ra]=cnt(l);y=tile(y,s);}

⍝ z.s[ra]=sum<B>(abs(x));if(!cnt(z)){z.v=scl(0);R;}

⍝ array w=where(x).as(s32);af::index ix[4];if(z.s[ra]==w.elements()){

⍝ ix[ra]=w;z.v=y(ix[0],ix[1],ix[2],ix[3]);R;}

⍝ array i=shift(accum(abs(x(w))),1),d=shift(w,1);i(0)=0;d(0)=0;

⍝ array v=array(z.s[ra],s32),u=array(z.s[ra],s32);v=0;u=0;

⍝ array s=(!sign(x(w))).as(s32);array t=shift(s,1);t(0)=0;

⍝ v(i)=w-d;u(i)=s-t;ix[ra]=accum(v);z.v=y(ix[0],ix[1],ix[2],ix[3]);

⍝ dim4 s1(1),s2(z.s);s1[ra]=z.s[ra];s2[ra]=1;u=array(accum(u),s1);

⍝ z.v\*=tile(u,(I)s2[0],(I)s2[1],(I)s2[2],(I)s2[3]);}

⍝ MF(rgt\_f){z=r;}

⍝ DF(rgt\_f){z=r;}

⍝ MF(rho\_f){I sp[4]={1,1,1,1};DO(r.r,sp[r.r-(i+1)]=(I)r.s[i]);

⍝ z.s=dim4(r.r);z.r=1;if(!cnt(z)){z.v=scl(0);R;}z.v=array(z.s,sp);}

⍝ DF(rho\_f){B cr=cnt(r);B cl=cnt(l);B s[4];if(l.r>1)err(11);if(cl>4)err(16);

⍝ l.v.as(s64).host(s);z.r=(I)cl;DO(4,z.s[i]=i>=z.r?1:s[z.r-(i+1)])B cz=cnt(z);

⍝ if(!cz){z.v=scl(0);R;}z.v=array(cz==cr?r.v:flat(r.v)(iota(cz)%cr),z.s);}

⍝ MF(rol\_f){z.r=r.r;z.s=r.s;if(!cnt(r)){z.v=r.v;R;}

⍝ array rnd=randu(r.v.dims(),f64);z.v=(0==r.v)\*rnd+trunc(r.v\*rnd);}

⍝

⍝ DF(rol\_f){if(cnt(r)!=1||cnt(l)!=1)err(5);

⍝ D lv=l.v.as(f64).scalar<D>();D rv=r.v.as(f64).scalar<D>();

⍝ if(lv>rv||lv!=floor(lv)||rv!=floor(rv)||lv<0||rv<0)err(11);

⍝ I s=(I)lv;I t=(I)rv;z.r=1;z.s=dim4(s);if(!s){z.v=scl(0);R;}

⍝ std::vector<I> g(t);std::vector<I> d(t);

⍝ ((1+range(t))\*randu(t)).as(s32).host(g.data());

⍝ DO(t,I j=g[i];if(i!=j)d[i]=d[j];d[j]=i)z.v=array(z.s,d.data());}

⍝ MF(rot\_f){z.r=r.r;z.s=r.s;z.v=flip(r.v,0);}

⍝ DF(rot\_f){I lc=(I)cnt(l);if(lc==1){z.r=r.r;z.s=r.s;

⍝ z.v=shift(r.v,-l.v.as(s32).scalar<I>());R;}

⍝ if(l.r!=r.r-1)err(5);DO(l.r,if(l.s[i]!=r.s[i+1])err(5))

⍝ std::vector<I> x(lc);l.v.as(s32).host(x.data());

⍝ z.v=array(r.v,r.s[0],lc);z.r=r.r;z.s=r.s;

⍝ DO(lc,z.v(span,i)=shift(z.v(span,i),-x[i]))z.v=array(z.v,z.s);}

⍝ MF(rtf\_f){z.r=r.r;z.s=r.s;z.v=r.r?flip(r.v,r.r-1):r.v;}

⍝ DF(rtf\_f){I lc=(I)cnt(l);I ra=r.r?r.r-1:0;I ix[]={0,0,0,0};

⍝ if(lc==1){z.r=r.r;z.s=r.s;ix[ra]=-l.v.as(s32).scalar<I>();

⍝ z.v=shift(r.v,ix[0],ix[1],ix[2],ix[3]);R;}

⍝ if(l.r!=r.r-1)err(5);DO(l.r,if(l.s[i]!=r.s[i])err(5))

⍝ std::vector<I> x(lc);l.v.as(s32).host(x.data());

⍝ z.v=array(r.v,lc,r.s[ra]);z.r=r.r;z.s=r.s;

⍝ DO(lc,z.v(i,span)=shift(z.v(i,span),0,-x[i]))

⍝ z.v=array(z.v,z.s);}

⍝ DF(scn\_f){if(r.s[0]!=1&&r.s[0]!=sum<I>(l.v>0))err(5);

⍝ if(l.r>1)err(5);array ca=max(1,abs(l.v)).as(s32);I c=sum<I>(ca);

⍝ if(!cnt(l))c=0;A t(r.r?r.r:1,r.s,scl(0));t.s[0]=c;

⍝ if(!cnt(t)){z=t;R;}t.v=array(t.s,r.v.type());t.v=0;

⍝ array pw=0<l.v;array pa=pw\*l.v;I pc=sum<I>(pa);if(!pc){z=t;R;}

⍝ pw=where(pw);pa=scan(pa(pw),0,AF\_BINARY\_ADD,false);

⍝ array si(pc,s32);si=0;si(pa)=1;si=accum(si)-1;

⍝ array ti(pc,s32);ti=1;ti(pa)=scan(ca,0,AF\_BINARY\_ADD,false)(pw);

⍝ ti=scanByKey(si,ti);t.v(ti,span)=r.v(si,span);z=t;}

⍝ DF(scf\_f){I ra=r.r?r.r-1:0;af::index sx[4];af::index tx[4];

⍝ if(r.s[ra]!=1&&r.s[ra]!=sum<I>(l.v>0))err(5);

⍝ if(l.r>1)err(5);array ca=max(1,abs(l.v)).as(s32);I c=sum<I>(ca);

⍝ if(!cnt(l))c=0;A t(ra+1,r.s,scl(0));t.s[ra]=c;

⍝ if(!cnt(t)){z=t;R;}t.v=array(t.s,r.v.type());t.v=0;

⍝ array pw=0<l.v;array pa=pw\*l.v;I pc=sum<I>(pa);if(!pc){z=t;R;}

⍝ pw=where(pw);pa=scan(pa(pw),0,AF\_BINARY\_ADD,false);

⍝ array si(pc,s32);si=0;si(pa)=1;si=accum(si)-1;sx[ra]=si;

⍝ array ti(pc,s32);ti=1;ti(pa)=scan(ca,0,AF\_BINARY\_ADD,false)(pw);

⍝ ti=scanByKey(si,ti);tx[ra]=ti;

⍝ t.v(tx[0],tx[1],tx[2],tx[3])=r.v(sx[0],sx[1],sx[2],sx[3]);z=t;}

⍝ MF(sqd\_f){z=r;}

⍝ DF(sqd\_f){if(l.r>1)err(4);B s=!l.r?1:l.s[l.r-1];

⍝ if(s>r.r)err(5);if(!cnt(l)){z=r;R;}

⍝ I sv[4];af::index x[4];l.v.as(s32).host(sv);

⍝ DO((I)s,if(sv[i]<0||sv[i]>=r.s[i])err(3));

⍝ DO((I)s,x[r.r-(i+1)]=sv[i]);z.r=r.r-(U)s;z.s=dim4(z.r,r.s.get());

⍝ z.v=r.v(x[0],x[1],x[2],x[3]);}

⍝ MF(sub\_f){z.r=r.r;z.s=r.s;z.v=-r.v;}

⍝ SF(sub\_f,z.v=lv-rv)

⍝ MF(tke\_f){z=r;}

⍝

⍝ DF(tke\_f){I lv[4];seq it[4];seq ix[4];B c=cnt(l);

⍝ if(l.r>1||(c>r.r&&r.r))err(4);if(!c){z=r;R;}

⍝ U rk=r.r?r.r:(U)l.s[0];z.r=rk;z.s=r.s;l.v.as(s32).host(lv);

⍝ DO((I)c,{U j=rk-(i+1);I a=std::abs(lv[i]);z.s[j]=a;

⍝ if(a>r.s[j])ix[j]=seq((D)r.s[j]);

⍝ else if(lv[i]<0)ix[j]=seq((D)r.s[j]-a,(D)r.s[j]-1);

⍝ else ix[j]=seq(a);

⍝ it[j]=ix[j]+(lv[i]<0)\*(a-(D)r.s[j]);})

⍝ if(!cnt(z)){z.v=scl(0);R;}z.v=array(z.s,r.v.type());z.v=0;

⍝ z.v(it[0],it[1],it[2],it[3])=r.v(ix[0],ix[1],ix[2],ix[3]);}

⍝ MF(trn\_f){z.r=r.r;DO(r.r,z.s[i]=r.s[r.r-(i+1)])

⍝ switch(r.r){CS(0,z.v=r.v)CS(1,z.v=r.v)CS(2,z.v=r.v.T())

⍝ CS(3,z.v=reorder(r.v,2,1,0))CS(4,z.v=reorder(r.v,3,2,1,0))}}

⍝ DF(trn\_f){I lv[4];if(l.r>1||cnt(l)!=r.r)err(5);

⍝ l.v.as(s32).host(lv);DO(r.r,if(lv[i]<0||lv[i]>=r.r)err(4))

⍝ U8 f[]={0,0,0,0};DO(r.r,f[lv[i]]=1)

⍝ U8 t=1;DO(r.r,if(t&&!f[i])t=0;else if(!t&&f[i])err(5))

⍝ DO(r.r,if(!f[i])err(16))

⍝ z.r=r.r;DO(r.r,z.s[r.r-(lv[i]+1)]=r.s[r.r-(i+1)])

⍝ I s[4];DO(r.r,s[r.r-(lv[i]+1)]=r.r-(i+1))

⍝ switch(r.r){CS(0,z.v=r.v)CS(1,z.v=r.v)

⍝ CS(2,z.v=reorder(r.v,s[0],s[1]))

⍝ CS(3,z.v=reorder(r.v,s[0],s[1],s[2]))

⍝ CS(4,z.v=reorder(r.v,s[0],s[1],s[2],s[3]))}}

⍝ MF(unq\_f){if(r.r>1)err(4);z.r=1;if(!cnt(r)){z.s=r.s;z.v=r.v;R;}

⍝ array a,b;sort(a,b,r.v);z.v=a!=shift(a,1);z.v(0)=1;

⍝ z.v=where(z.v);sort(b,z.v,b(z.v),a(z.v));

⍝ z.s=dim4(z.v.elements());}

⍝ DF(unq\_f){if(r.r>1||l.r>1)err(4);z.r=1;dtype mt=mxt(l.v,r.v);

⍝ if(!cnt(l)){z.s=r.s;z.v=r.v;R;}if(!cnt(r)){z.s=l.s;z.v=l.v;R;}

⍝ array x=setUnique(l.v);B c=x.elements();

⍝ z.v=!anyTrue(tile(r.v,1,(U)c)==tile(array(x,1,c),(U)r.s[0],1),1);

⍝ z.v=join(0,l.v.as(mt),r.v(where(z.v)).as(mt));

⍝ z.s=dim4(z.v.elements());}

⍝

⍝ #define brkop(zz,ll,rr,pp) brk\_o zz(ll,rr,pp)

⍝ #define comop(zz,rr,pp) com\_o zz(rr,pp)

⍝ #define dotop(zz,ll,rr,pp) dot\_o zz(ll,rr,pp)

⍝ #define mapop(zz,rr,pp) map\_o zz(rr,pp)

⍝ #define redop(zz,rr,pp) red\_o zz(rr,pp)

⍝ #define jotop(zz,ll,rr,pp) jot\_o zz(ll,rr,pp)

⍝ #define oupop(zz,rr,pp) oup\_o zz(rr,pp)

⍝ #define powop(zz,ll,rr,pp) pow\_o zz(ll,rr,pp)

⍝ #define rdfop(zz,rr,pp) rdf\_o zz(rr,pp)

⍝ #define rnkop(zz,ll,rr,pp) rnk\_o zz(ll,rr,pp)

⍝ #define scnop(zz,rr,pp) scn\_o zz(rr,pp)

⍝ #define scfop(zz,rr,pp) scf\_o zz(rr,pp)

⍝

⍝ MF(brk\_o){ll(z,r,(r.r?r.r-1:0)-ww.v.as(f64).scalar<D>(),p);}

⍝ DF(brk\_o){D ax=l.r;if(r.r>l.r)ax=r.r;if(ax)ax--;

⍝ ll(z,l,r,ax-ww.v.as(f64).scalar<D>(),p);}

⍝ MF(com\_o){ll(z,r,r,p);}DF(com\_o){ll(z,r,l,p);}

⍝ DF(dot\_o){I ra=r.r?r.r-1:0;if(r.r&&l.r&&l.s[0]!=r.s[ra])err(5);

⍝ I la=l.r?l.r-1:0;A t(la+ra,r.s,r.v(0));if(t.r>4)err(10);

⍝ t.s[ra]=1;DO(la,t.s[i+ra]=l.s[i+1])if(!cnt(t)){t.v=scl(0);z=t;R;}

⍝ if(!l.s[0]||!r.s[ra]){t.v=ll.id(t.s);z=t;R;}

⍝ I c=(I)(l.r?l.s[0]:r.s[ra]);

⍝ I rc=(I)(cnt(r)/r.s[ra]);I lc=(I)(cnt(l)/l.s[0]);

⍝ array x=array(l.v,(I)l.s[0],lc);array y=array(r.v,rc,(I)r.s[ra]);

⍝ if(1==l.s[0]){x=tile(x,c,1);}if(1==r.s[ra]){y=tile(y,1,c);}

⍝ if("add"==ll.nm&&"mul"==rr.nm){

⍝ t.v=array(matmul(y.as(f64),x.as(f64)),t.s);z=t;R;}

⍝ x=tile(array(x,c,1,lc),1,rc,1);y=tile(y.T(),1,1,lc);

⍝ A X(3,dim4(c,rc,lc),x.as(f64));A Y(3,dim4(c,rc,lc),y.as(f64));

⍝ mapop(mfn,rr,p);redop(rfn,ll,p);mfn(X,X,Y,p);rfn(X,X,p);

⍝ t.v=array(X.v,t.s);z=t;}

⍝ MF(jot\_o){if(!fl){rr(z,aa,r,p);R;}if(!fr){ll(z,r,ww,p);R;}

⍝ rr(z,r,p);ll(z,z,p);}

⍝ DF(jot\_o){if(!fl||!fr){err(2);}rr(z,r,p);ll(z,l,z,p);}

⍝ MF(map\_o){if(scm(ll)){ll(z,r,p);R;}

⍝ z.r=r.r;z.s=r.s;I c=(I)cnt(z);if(!c){z.v=scl(0);R;}

⍝ A zs;A rs=scl(r.v(0));ll(zs,rs,p);if(c==1){z.v=zs.v;R;}

⍝ array v=array(z.s,zs.v.type());v(0)=zs.v(0);

⍝ DO(c-1,rs.v=r.v(i+1);ll(zs,rs,p);v(i+1)=zs.v(0))z.v=v;}

⍝ DF(map\_o){if(scd(ll)){ll(z,l,r,p);R;}

⍝ if((l.r==r.r&&l.s==r.s)||!l.r){z.r=r.r;z.s=r.s;}

⍝ else if(!r.r){z.r=l.r;z.s=l.s;}else if(l.r!=r.r)err(4);

⍝ else if(l.s!=r.s)err(5);else err(99);I c=(I)cnt(z);

⍝ if(!c){z.v=scl(0);R;}A zs;A rs=scl(r.v(0));A ls=scl(l.v(0));

⍝ ll(zs,ls,rs,p);if(c==1){z.v=zs.v;R;}

⍝ array v=array(z.s,zs.v.type());v(0)=zs.v(0);

⍝ if(!r.r){rs.v=r.v;

⍝ DO(c-1,ls.v=l.v(i+1);ll(zs,ls,rs,p);v(i+1)=zs.v(0);)

⍝ z.v=v;R;}

⍝ if(!l.r){ls.v=l.v;

⍝ DO(c-1,rs.v=r.v(i+1);ll(zs,ls,rs,p);v(i+1)=zs.v(0);)

⍝ z.v=v;R;}

⍝ DO(c-1,ls.v=l.v(i+1);rs.v=r.v(i+1);ll(zs,ls,rs,p);

⍝ v(i+1)=zs.v(0))z.v=v;}

⍝ DF(oup\_o){A t(l.r+r.r,r.s,r.v(0));if(t.r>4)err(10);

⍝ DO(l.r,t.s[i+r.r]=l.s[i])if(!cnt(t)){t.v=scl(0);z=t;R;}

⍝ array x(flat(l.v),1,cnt(l));array y(flat(r.v),cnt(r),1);

⍝ dim4 ts(cnt(r),cnt(l));x=tile(x,(I)ts[0],1);y=tile(y,1,(I)ts[1]);

⍝ mapop(mfn,ll,p);A xa(2,ts,x);A ya(2,ts,y);mfn(xa,xa,ya,p);

⍝ t.v=array(xa.v,t.s);z=t;}

⍝ MF(pow\_o){if(fr){A t;A v=r;

⍝ do{A u;ll(u,v,p);rr(t,u,v,p);if(t.r)err(5);v=u;}

⍝ while(!t.v.as(s32).scalar<I>());z=v;R;}

⍝ if(ww.r)err(4);I c=ww.v.as(s32).scalar<I>();z=r;DO(c,ll(z,z,p))}

⍝ DF(pow\_o){if(fr){A t;A v=r;

⍝ do{A u;ll(u,l,v,p);rr(t,u,v,p);if(t.r)err(5);v=u;}

⍝ while(!t.v.as(s32).scalar<I>());z=v;R;}

⍝ if(ww.r)err(4);I c=ww.v.as(s32).scalar<I>();

⍝ A t=r;DO(c,ll(t,l,t,p))z=t;}

⍝ MF(rdf\_o){A t(r.r?r.r-1:0,dim4(1),r.v(0));DO(t.r,t.s[i]=r.s[i])

⍝ I rc=(I)r.s[t.r];I zc=(I)cnt(t);mapop(mfn,ll,p);

⍝ if(!zc){t.v=scl(0);z=t;R;}if(!rc){t.v=ll.id(t.s);z=t;R;}

⍝ if(1==rc){t.v=array(r.v,t.s);z=t;R;}

⍝ if("add"==ll.nm){if(r.v.isbool())t.v=count(r.v,t.r).as(s32);

⍝ else t.v=sum(r.v.as(f64),t.r);z=t;R;}

⍝ if("mul"==ll.nm){t.v=product(r.v.as(f64),t.r);z=t;R;}

⍝ if("min"==ll.nm){t.v=min(r.v,t.r);z=t;R;}

⍝ if("max"==ll.nm){t.v=max(r.v,t.r);z=t;R;}

⍝ if("and"==ll.nm){t.v=allTrue(r.v,t.r);z=t;R;}

⍝ if("lor"==ll.nm){t.v=anyTrue(r.v,t.r);z=t;R;}

⍝ af::index x[4];x[t.r]=rc-1;t.v=r.v(x[0],x[1],x[2],x[3]);

⍝ DO(rc-1,x[t.r]=rc-(i+2);

⍝ mfn(t,A(t.r,t.s,r.v(x[0],x[1],x[2],x[3])),t,p));z=t;}

⍝ DF(rdf\_o){if(l.r!=0&&(l.r!=1||l.s[0]!=1))err(5);if(!r.r)err(4);

⍝ I lv=l.v.as(s32).scalar<I>();I ra=r.r-1;

⍝ if((r.s[ra]+1)<lv)err(5);I rc=(I)((1+r.s[ra])-abs(lv));

⍝ mapop(mfn,ll,p);A t(r.r,r.s,scl(0));t.s[ra]=rc;if(!cnt(t)){z=t;R;}

⍝ if(!lv){t.v=ll.id(t.s);z=t;R;}seq rng(rc);af::index x[4];

⍝ if(lv>=0){x[ra]=rng+((D)lv-1);t.v=r.v(x[0],x[1],x[2],x[3]);

⍝ DO(lv-1,x[ra]=rng+((D)lv-(i+2));

⍝ mfn(t,A(t.r,t.s,r.v(x[0],x[1],x[2],x[3])),t,p))

⍝ }else{x[ra]=rng;t.v=r.v(x[0],x[1],x[2],x[3]);

⍝ DO(abs(lv)-1,x[ra]=rng+(D)(i+1);

⍝ mfn(t,A(t.r,t.s,r.v(x[0],x[1],x[2],x[3])),t,p))}

⍝ z=t;}

⍝ MF(red\_o){A t(r.r?r.r-1:0,dim4(1),z.v);DO(t.r,t.s[i]=r.s[i+1])

⍝ I rc=(I)r.s[0];I zc=(I)cnt(t);if(!zc){t.v=scl(0);z=t;R;}

⍝ if(!rc){t.v=ll.id(t.s);z=t;R;}

⍝ if(1==rc){t.v=array(r.v,t.s);z=t;R;}

⍝ if("add"==ll.nm){if(r.v.isbool())t.v=count(r.v,0).as(s32);

⍝ else t.v=sum(r.v.as(f64),0);z=t;R;}

⍝ if("mul"==ll.nm){t.v=product(r.v.as(f64),0);z=t;R;}

⍝ if("min"==ll.nm){t.v=min(r.v,0);z=t;R;}

⍝ if("max"==ll.nm){t.v=max(r.v,0);z=t;R;}

⍝ if("and"==ll.nm){t.v=allTrue(r.v,0);z=t;R;}

⍝ if("lor"==ll.nm){t.v=anyTrue(r.v,0);z=t;R;}

⍝ t.v=r.v(rc-1,span);mapop(mfn,ll,p);

⍝ DO(rc-1,mfn(t,A(t.r,t.s,r.v(rc-(i+2),span)),t,p))z=t;}

⍝ DF(red\_o){if(l.r!=0&&(l.r!=1||l.s[0]!=1))err(5);if(!r.r)err(4);

⍝ I lv=l.v.as(s32).scalar<I>();if((r.s[0]+1)<lv)err(5);

⍝ I rc=(I)((1+r.s[0])-abs(lv));mapop(mfn,ll,p);

⍝ A t(r.r,r.s,scl(0));t.s[0]=rc;if(!cnt(t)){z=t;R;}

⍝ if(!lv){t.v=ll.id(t.s);z=t;R;}seq rng(rc);

⍝ if(lv>=0){t.v=r.v(rng+((D)lv-1),span);

⍝ DO(lv-1,mfn(t,A(t.r,t.s,r.v(rng+((D)lv-(i+2)),span)),t,p))

⍝ }else{t.v=r.v(rng,span);

⍝ DO(abs(lv)-1,mfn(t,A(t.r,t.s,r.v(rng+(D)(i+1),span)),t,p))}

⍝ z=t;}

⍝

⍝ MF(rnk\_o){if(cnt(ww)!=1)err(4);I cr=ww.v.as(s32).scalar<I>();

⍝ if(scm(ll)||cr>=r.r){ll(z,r,p);R;}

⍝ if(cr<=-r.r||!cr){mapop(f,ll,p);f(z,r,p);R;}

⍝ if(cr<0)cr=r.r+cr;if(cr>3)err(10);I dr=r.r-cr;

⍝ dim4 sp(1);DO(dr,sp[cr]\*=r.s[i+cr])DO(cr,sp[i]=r.s[i])

⍝ std::vector<A> tv(sp[cr]);A b(cr+1,sp,array(r.v,sp));

⍝ DO((I)sp[cr],sqdfn(tv[i],scl(scl(i)),b,p);ll(tv[i],tv[i],p))

⍝ I mr=0;dim4 ms(1);dtype mt=b8;if(mr>3)err(10);

⍝ DO((I)sp[cr],if(mr<tv[i].r)mr=tv[i].r;mt=mxt(mt,tv[i]);I si=i;

⍝ DO(4,if(ms[3-i]<tv[si].s[3-i]){ms=tv[si].s;break;}))

⍝ I mc=(I)cnt(ms);array v(mc\*sp[cr],mt);v=0;

⍝ DO((I)sp[cr],seq ix((D)cnt(tv[i]));v(ix+(D)(i\*mc))=flat(tv[i].v))

⍝ z.r=mr+dr;z.s=ms;z.s[mr]=sp[cr];z.v=array(v,z.s);}

⍝ DF(rnk\_o){I cl,cr,dl,dr;dim4 sl(1),sr(1);array wwv=ww.v.as(s32);

⍝ if(cnt(ww)==1)cl=cr=wwv.scalar<I>();

⍝ else if(cnt(ww)==2){cl=wwv.scalar<I>();cr=wwv(1).scalar<I>();}

⍝ else err(4);

⍝ if(cl>l.r)cl=l.r;if(cr>r.r)cr=r.r;if(cl<-l.r)cl=0;if(cr<-r.r)cr=0;

⍝ if(cl<0)cl=l.r+cl;if(cr<0)cr=r.r+cr;if(cr>3||cl>3)err(10);

⍝ dl=l.r-cl;dr=r.r-cr;if(dl!=dr&&dl&&dr)err(4);

⍝ if(dl==dr)DO(dr,if(l.s[i+cl]!=r.s[i+cr])err(5))

⍝ DO(dl,sl[cl]\*=l.s[i+cl])DO(cl,sl[i]=l.s[i])

⍝ DO(dr,sr[cr]\*=r.s[i+cr])DO(cr,sr[i]=r.s[i])

⍝ B sz=dl>dr?sl[cl]:sr[cr];std::vector<A> tv(sz);

⍝ A a(cl+1,sl,array(l.v,sl));A b(cr+1,sr,array(r.v,sr));

⍝ I mr=0;dim4 ms(1);dtype mt=b8;

⍝ DO((I)sz,A ta;A tb;A ai=scl(scl(i%sl[cl]));A bi=scl(scl(i%sr[cr]));

⍝ sqdfn(ta,ai,a,p);sqdfn(tb,bi,b,p);ll(tv[i],ta,tb,p);

⍝ if(mr<tv[i].r)mr=tv[i].r;mt=mxt(mt,tv[i]);A t=tv[i];

⍝ DO(4,if(ms[i]<t.s[i])ms[i]=t.s[i]))

⍝ B mc=cnt(ms);array v(mc\*sz,mt);v=0;

⍝ DOB(sz,seq ix((D)cnt(tv[i]));v(ix+(D)(i\*mc))=flat(tv[i].v))

⍝ z.r=mr+(dr>dl?dr:dl);z.s=ms;z.s[mr]=sz;z.v=array(v,z.s);}

⍝ MF(scn\_o){z.r=r.r;z.s=r.s;I rc=(I)r.s[0];

⍝ if(1==rc){z.v=r.v;R;}if(!cnt(z)){z.v=scl(0);R;}

⍝ if("add"==ll.nm){z.v=scan(r.v.as(f64),0,AF\_BINARY\_ADD);R;}

⍝ if("mul"==ll.nm){z.v=scan(r.v.as(f64),0,AF\_BINARY\_MUL);R;}

⍝ if("min"==ll.nm){z.v=scan(r.v.as(f64),0,AF\_BINARY\_MIN);R;}

⍝ if("max"==ll.nm){z.v=scan(r.v.as(f64),0,AF\_BINARY\_MAX);R;}

⍝ mapop(mfn,ll,p);z.v=array(z.s,f64);A t(z.r?z.r-1:0,z.s,r.v(0));

⍝ DO(t.r,t.s[i]=t.s[i+1]);t.s[t.r]=1;I tc=(I)cnt(t);

⍝ DO(rc,t.v=r.v(i,span).as(f64);I c=i;

⍝ DO(c,mfn(t,A(t.r,t.s,r.v(c-(i+1),span)),t,p))

⍝ z.v(i,span)=t.v)}

⍝ MF(scf\_o){z.r=r.r;z.s=r.s;I ra=r.r?r.r-1:0;I rc=(I)r.s[ra];

⍝ if(1==rc){z.v=r.v;R;}if(!cnt(z)){z.v=scl(0);R;}

⍝ if("add"==ll.nm){z.v=scan(r.v.as(f64),ra,AF\_BINARY\_ADD);R;}

⍝ if("mul"==ll.nm){z.v=scan(r.v.as(f64),ra,AF\_BINARY\_MUL);R;}

⍝ if("min"==ll.nm){z.v=scan(r.v.as(f64),ra,AF\_BINARY\_MIN);R;}

⍝ if("max"==ll.nm){z.v=scan(r.v.as(f64),ra,AF\_BINARY\_MAX);R;}

⍝ z.v=array(z.s,f64);A t(z.r?z.r-1:0,z.s,r.v(0));t.s[ra]=1;

⍝ I tc=(I)cnt(t);af::index x[4];mapop(mfn,ll,p);

⍝ DO(rc,x[ra]=i;t.v=r.v(x[0],x[1],x[2],x[3]).as(f64);I c=i;

⍝ DO(c,x[ra]=c-(i+1);

⍝ mfn(t,A(t.r,t.s,r.v(x[0],x[1],x[2],x[3])),t,p))

⍝ x[ra]=i;z.v(x[0],x[1],x[2],x[3])=t.v)}

⍝

⍝ EXPORT A\*mkarray(lp\*d){A\*z=new A;cpda(\*z,d);R z;}

⍝ EXPORT V frea(A\*a){delete a;}

⍝ EXPORT V exarray(lp\*d,A\*a){cpad(d,\*a);}

⍝ EXPORT V afsync(){sync();}

⍝ EXPORT Window \*w\_new(char \*k){R new Window(k);}

⍝ EXPORT I w\_close(Window\*w){R w->close();}

⍝ EXPORT V w\_del(Window\*w){delete w;}

⍝ EXPORT V w\_img(lp\*d,Window\*w){A a;cpda(a,d);

⍝ w->image(a.v.as(a.r==2?f32:u8));}

⍝ EXPORT V w\_plot(lp\*d,Window\*w){A a;cpda(a,d);w->plot(a.v.as(f32));}

⍝ EXPORT V w\_hist(lp\*d,D l,D h,Window\*w){A a;cpda(a,d);

⍝ w->hist(a.v.as(u32),l,h);}

⍝ EXPORT V loadimg(lp\*z,char\*p,I c){array a=loadImage(p,c);

⍝ A b(a.numdims(),a.dims(),a.as(s16));cpad(z,b);}

⍝ EXPORT V saveimg(lp\*im,char\*p){A a;cpda(a,im);

⍝ saveImageNative(p,a.v.as(a.v.type()==s32?u16:u8));}

:EndNamespace