THE G15 PMN FCM Spreadsheet Source Code

by S.R. Weber

Copyright (c) S.R. Weber, 2017. Reproduction of the whole program acceptable on condition that, as is common with open source programs, due acknowledgement is given, incl reference to source location, which is: norskesites.org/fic3/fic3inf3.htm

Published January 20th, 2017; contains references to the Third Foundation which was finalised December 15th, 2015. G15 PMN is a programming language. More info about it and its FCM approach to structuring programming at the above link.

```
<h5415>
|Spreadsheet   |By S_R_Weber;
|calculation   |G15 pmn fcm
|screen with   |is used and
|easy export   |standard 3rd
|of document   |foundation
|to b9edit     |with one new
|app#:         |extension,bt,
|3555558       |b9font toning

<h5416>
|This g15 pmn  |regards
|fcm program   |program
|is also a     |structure;
|kind of       |for pattern
|tutorial of   |matching &
|advanced      |task steering
|programming   |cfr g15 pmn
|in fcm, as    |fcm robotapps
```

```
<h5417>
|This fcm      |program, and
|program       |it will have
|organises fcm |an ease as
|foundries in  |regards the
|ways showing  |thinking
|how you can   |about it
|do every      |beyond other
|type of       |approaches

<h5418>
|The toplevel  |positions in
|foundry has   |spreadsheet
|the keyboard  |links to;when
|and screen    |this is put
|handling; it  |to '1', all
|links to a    |recalculation
|'refreshfn'   |takes place
|which also    |in positions

<h5419>
|Each pos has  |it also has
|links to      |an fnact that
|two pairs of  |can run thru
|nodes holding |this formula
|arraydata--   |array & call
|text & the    |each foundry
|'compiled     |referred to
|formula';     |in it

<h5420>
|Each operator |such foundry
|in formula is |we set it to
|'compiled'    |passive &
|into the      |rather use
|number of the |routines like
|foundry--by   |'performfn'
|fnam--and to  |to start them
|call on any   |as needed

<h5421>
|A clue to     |few variables
|good fcm      |are here used
|structure is  |except local
|keep all data |to some sub-
|{when speed   |routines of
|acceptable}   |fnacts, which
|inside the    |drive all
|nodes; thus,  |events here

<h5422>
|In the start  |structures
|of this fcm   |are included,
|application   |on top of the
|several       |3rd found;
|useful extra  |these are
|definitions   |general & can
|for numbers,  |be called by
|texts and fcm |fnacts

<h5423>
|Compared to   |package, as
|the super-    |finalized
|model theory  |2016::12::15,
|physics       |you find here
|examples in   |a much more
|the standard  |polished
|third         |and two-way
|foundation    |interface;

<h5424>
|Since a       |solution is
|spreadsheet   |to let the
|can have      |fcm loop
|formulae      |'pulse' to
|which goes    |put numbers
|in both       |thru in all
|directions,   |directions, &
|the elegant   |ctr-t toggle;

<h5425>
|In the        |many more
|physics       |kinds; you
|examples,     |can build on
|the node use  |this fcm
|is of one     |framework for
|kind; here,   |an enormous
|we find       |variety of
|hints of      |applications!

<h5426>
|After careful |For loops are
|meditation    |generally the
|over it, the  |most dramatic
|exact loop    |action by a
|structures    |pc, and so
|defined in tf |most in need
|are kept with |of structural
|nothing added |expliciteness

<h5427>
|All through   |terminal
|the making of |provided
|this program  |lively checks
|over the 3    |of all parts
|weeks it      |of program,
|took, the     |as it grew,
|interactive   |step by step,
|g15 pmn       |naturally

<h5428>
|First-hand    |readable form
|relation to   |from car edit
|data in this  |   For more
|application's |fcm tuturial
|emphasized by |comments, pls
|putting data  |fine-read all
|to cards in   |the cards
|a fairly      |that follow

<h5429>
columnspace=   |export to
168.           |b9edit;
               |As you rework
|This          |this program
|spreadsheet   |into another,
|is tailormade |'columnspace'
|to provide    |may have to
|seamless      |be changed

<h5430>
advancedckms=  |it can be
100.           |changed
               |{ie, it can
|This is the   |be thought of
|qty given to  |as a 'global
|activepause   |setting'} by
|when you use  |the function
|advancedck    |setfastvar

<h5431>
advancedck=    1
|gives:flag    ck
|as ck, it
|checks        se
|keyboard,
|but allows
|a little
|time as well  ex

<h5432>
advancedckms   sh







activepause    ck.
```

```
<h5433>
extratriplet=  i1
|in:a b c      i2
|gives:a b c   i3
|a b c

s3             i1
s2             i2
s1             i3.

<h5434>
sorttwo=       d
|in:a b        d
|gives:x y     gt
|action:
|highest of    se
|the two are
|top of
|stack         w.

<h5435>
filtercharnum= |instance of
|In:n, source, |asciinum
|destination   |given {eg 122
|Action:source |for z} is
|quote is      |removed; len
|copied over   |updated; be
|to dest quote |sure dest is
|where any     |roomy enough

<h5436>
sx             d4
tx
s5
jx
lk             basis
s9             ix
i9             setlenandnil
ye             ex

<h5437>
1              i1
s4             jx
               ay




ll:2000000000  s6

<h5438>
i5             i6
```

```
i6             i4
eq             ix

               ya


d5             h4

<h5439>
i9             i4
i1             dc
gt             ix
               setlenandnil
               ex


d5             lo.

<h5440>
tento=         sx
|In:quantity
|Gives:number
|Action:gives
|ten raised to
|quantity;must
|be prechecked
|range {0->9}  1

<h5441>
ll:2000000000  10
i1             mm
ix             |Loop is fast
gt             |and so
               |presumes eg
se             |'makefit'
               |as for input
ex             lo.

<h5442>
deleteachar=   tx
|In:pos, quote |pos is 1->n;
|Action:       |it exits
|deletes char  |cleanly when
|in quote and  |pos<1 or >n;
|updates       |Complements
|length and    |insertachar
|nilchar       s5

<h5443>
jx             isnotwithin
lk
s9             se
```

```
               ex
i5             !1
1              jx
i9             ku

<h5444>
|From:         |Qty:
jx             i9
i5             i5
ad             su
up             up
|To:           |including
f              |nilchar
dc             fw.

<h5445>
findcharnum=   tx
|In:chnum,text |chnum is
|Gives: pos    |ascii, eg
|Action: tells |97 for a;
|position 1->n |this handles
|of first      |also
|occurrence of |basislen
|charnum, or 0 s5

<h5446>
jx             d2
lk
s9



i9             basis
ye             ex

<h5447>
i5             s8
i5             i8
               ye

jx             d2
up
i9             basis
aw             ex

<h5448>
i8
jx
```

```
su.

<h5449>
rightaligntx1= rightaligntx1
^.




235
sz
&&             kl

<h5450>
rightalign=    i3
|In:newlen,txt lk
|Gives:newtxt  s2
|Action:Adds   |len:i2 new:i1
|startspaces;  rightaligntx1
|newlenmax:232 lk
s3             f
s1             s4

<h5451>
i1             i1
i4             iswithin
ns             n?
i1             |txtlen mustnt
i4             |exceed newlen
setlenandnil   se
i2             |nilstr ok
1              ex

<h5452>
|From:         ad
i3             up
up
|To:           |Qty:
i1             i2
i2
su
i4             fw.

<h5453>
fixsign=       |In other
|In:quote      |words, use
|Action:the    |makenumber
|makenumber    |then this
|uses ! for    |after when
|signed nums;  |a program
|use this to   |wants dash
|make it dash  |instead

<h5454>
```

```
up              se
t7
                ex
j7
lk
33              45
eq              j7
n?              kl.

<h5455>
|Using some of |of 32-bit
|the previous  |functions to
|functions, &  |parse, calc,
|culimating in |& format nums
|esp fetchnum  |{not 'undef'}
|formatnum,    |numgroupsign
|decimalmm etc |and/or
|next is a set |'decimalsign'

<h5456>
decimalsign=   togglenumsign=
46.            ^decimalsign
|Ascii dot:46  tx
|Ascii ',':44  ^numgroupsign
numgroupsign=  sx
44.            decimalsign
|very fast     numgroupsign
|global vars   |read on:>>

<h5457>
jx             |While USE of
setfastvar     |them is fast,
ix             |dot/comma
setfastvar.    |setting is
|For more      |typically a
|rapid change  |"global
|of them, put  |setting",&not
|warps to vars |done in loops

<h5458>
makenumr14tx=  makenumr14tx
^.




14
sz
&&             kl

<h5459>
makenumr14=    makenumber
|In:number     s3
|Gives:quote   |This is for a
```

```
|Action:right  |whole number,
|aligns        |no groupsigns
|number;       makenumr14tx
|employs '-'   lk
|when <0       s1

<h5460>
45             d3

               f
f3             f3
lk             kl
33
eq
n?             sh

<h5461>
14             |From:
i1             i3





ns             up

<h5462>
|To:           |Qty:
15             i3
i3             lk
lk             fw
su             14
i1             i1
               setlenandnil
ad             i1.

<h5463>
numberlike=    s1
|In:quote      |whole numbers
|Gives:flag    |only;
|Action:checks |not for dots
|whether at    |or commas;
|least         |dash ok
|BEGINNING is  i1
|numberlike    lk

<h5464>
ye             f1
               lk
               s3
d2
               f1
               up
```

```
basis          lk
ex             s5

<h5465>
i3             |These funcs
48             |are for
57             |economical
iswithin       |work etc;num
i3             |'undefined'
45             |is relevant
eq             |mostly in
or             |'tech' progs

<h5466>
i5             or
48
57
iswithin


i5
n?             an.

<h5467>
getnumlike=    |1, it also
|In:quote      |calls intonum
|Gives:flag,   |{at flag
|number        |basis it
|Action:same   |gives basis
|as numberlike |as num}
|in flag, but
|when flag is  tx

<h5468>
jx             basis
numberlike     basis
|Note: a text  ex
|with digits
|first, then
|chars, gives  dance
|flag 1, num 0 jx
d3             intonum.

<h5469>
leisurenum=    s1
|In:quote      |both dots and
|Gives:flag    |commas can be
|Action:loose  |at 2nd pos;
|check as to   |dash can be
|whether       |at first pos
|BEGINNING     i1
|could be num; lk
```

```
<h5470>
ye             f1
               lk
               s3
d2
               f1
               up
basis          lk
ex             s5

<h5471>
i3
48
57
iswithin
i3
45
eq
or

<h5472>
i5             or
48
57
iswithin

decimalsign
i5
eq

<h5473>
numgroupsign   or
i5
eq
or


i5
n?             an.

<h5474>
digitfillstx1= digitfillstx1
^.             |pls call them
|For these     |each with a
|number        |quote that
|routines,     |isn't the
35             |output of the
sz             |SAME func :-)
&&             kl

<h5475>
digitfills=    45
|In:n,numtxt   t5
```

```
|Gives:newtxt  |Eg, with n=2:
|n=qtydecimals |"1"   >"001"
|or 0;spaceles |"-12" >"-012"
|numtxt,can've |"1234">"1234"
|dash, gets    |""    >"000";
|enough digits |no groupsign

<h5476>
tx             lk
s3             s2
digitfillstx1  jx
lk             up
f              lk
|txt at stk    j5
sx             eq
jx             t6

<h5477>
f3             |Ascii basis:
|At least 1    48
|digit pre dot i1
j6             ix
ad
i2             fc
maxofthis      j6
s1             n?

<h5478>
d4             i1
               ix
|sign first    setlenandnil
               |Qty digits:
j5             i2
ix             j6
up             su
kl             s8

<h5479>
|from:         ad
jx             i8
up
j6             su
ad             up
|to:           |qty:
ix             i8
i1             fw.

<h5480>
indecimalsign= |must've space
|In:n, quote   |Normally, use
|Action:       |'digitfills'
|{handles n    |1st so for
|basis}; n pos |sure num has
```

```
|from right    |digits BOTH
|'decimalsign' |left & right
|put in; quote |of position n

<h5481>
tx             |That is to
s5             |say, whole
               |numbers are
i5             |correctly
n?             |handled by
               |this

se             ex

<h5482>
& &            i4
s4             jx
               jx
               lk
decimalsign    i5
i4             su
up             up
kl             insertachar.

<h5483>
innumgrouptx1= innumgrouptx1
^.
|Ample size    |Textspace for
               |commagrouped
               |number for
35             |both next
sz             |routines
&&             kl

<h5484>
winnumgroup=   s4
|In:numquote1  |Must be an ok
|Gives:        |number, can
|numquote2     |be &0& but
|Action: for a |not &&; no
|Whole num,can |xtra chars;
|have '-',puts |Subroutine in
|in groupsigns |'innumgroup'

<h5485>
i4             j2
lk             eq
s5             t3
45             innumgrouptx1
t2             lk
i4             |stk:result
up             f
lk             s1
```

```
<h5486>
j3             h1
n?             j2
               i1
               kl
               h4
               i5
               dc
d8             s5

<h5487>
|Calc new len:
i5
dc
3              |Acceptable
di             |input eg &1&
i5             |but not &&
ad
s3

<h5488>
|At stk:result i1
               sx

i3
j3
ad
d              i5
setlenandnil   s2

<h5489>
ll:2000000000  q3

i2             i3
i4             n?
wk
i3             se
ix
kw             ex

<h5490>
q2             d5
i5
i2             numgroupsign
su             i3
3              ix
mo             ya
               q3
ye             lo.

<h5491>
innumgroup=    tx
```

```
|In:n1, txt1   s1
|Gives: txt2   |{n1 can be 0}
|Puts fastvar  |txt1len > 0:
|numgroupsign  |at least 1
|to decimal    |digits before
|or whole num; |& n1 digits
|n1=qtydecimal |after decsign

<h5492>
i1             jx
ye             winnumgroup
|Note, in any  |can be signed
|func using fw |num with dash
|top important |Decimalsign
|to give the   |is fastvar
|right input   |'decimalsign'
d3             ex

<h5493>
|When qty of
|decimals is   |w/qtydecimals
|eg 2, then    |2 & dot=",",
|least input   |then "0,69"ok
|len is 4:     |also "-0,69",
|two decimals, |but not ",69"
|one sign, &   |In any case,
|one digit     |prechk input!

<h5494>
|Give the jx   i2
|len that      i1
|fits          up
|winnumgroup:  su
jx             f
lk             s3
               jx
s2             kl

<h5495>
|Call
|winnumgroup:
jx
winnumgroup
|for this
|uses initial
|leninfo only
sx

<h5496>
|Copy decimals lk
|From:         ix
jx             ad
i3             up
```

```
ad             |Qty:
up             f1
|To:
ix             fw

<h5497>
|Finally,      ix
|assert new    setlenandnil
|length for
|result
f1
ix
lk
ad             ix.

<h5498>
scaleint=      |decimals so
|In:n1,n2,n3   |it gets n2
|Gives:n4      |decimals, by
|Action:scales |rounding&div
|a whole, ie,  |or by mm w/10
|integer, num, |n2 or n3 can
|n3, imagined  |be basis;can
|to have n1    |be equal,too

<h5499>
s3             d2
s2
s1

i1
i2
eq             i3
n?             ex

<h5500>
i1             i3
i2             i1
le             i2
               su
               tn
               tento
               rd
d8             ex

<h5501>
|Scaleint is   i3
|good when     i2
|internal nums i1
|w/uneq decim  su
|qty are to be
|put into eg   tento
|decimalmm
```

```
|or decimalrd  mm.

<h5502>
decimalscale=  |w/a number,
|In:n1 txt     |with/without
|Gives:number  |'decimalsign'
|Action: where |no spaces or
|n1 is qty of  |groupsigns,
|DESIRED       |gives rightly
|decimals, &   |scaled num;
|txt a quote   |CHANGES TXT!

<h5503>
tx             |qty decimals
sx             |eg set to 2:
|Rounds number |"1234567,89":
|when too many |123456789
|decimals;len  |"-69,9099":
|can be 0; eg, |-6991
|w/decimalsign |"2323":232300
|set to "," &  |and "0,0":0

<h5504>
jx             basis
lk
s1             |This is
               |subroutine
i1             |in fetchnum
ye

d2             ex

<h5505>
decimalsign    jx
jx             intonum
findcharnum    t8
s4
i4             i4
jx             ye
|handles basis
deleteachar    d5

<h5506>
basis          i1
ix             i4
j8
               su
               ix
scaleint
               j8
ex             scaleint.

<h5507>
```

```
fetchnumtx1=   fetchnumtx1
^.




35
sz
&&             kl

<h5508>
fetchnum=      |Txt must have
|In:n1,n2,txt  |no spaces;can
|Gives:number  |have dash;can
|Action:n1 is  |always have
|flag"is group |'decimalsign'
|signs ok?";n2 |Result scaled
|is basis or   |Note:txt may
|qty decimals; |be changed!

<h5509>
tx             numgroupsign
s4             jx
s5             fetchnumtx1
|Avoid 'undef' lk
               f
i5             tx
n?
d7             filtercharnum

<h5510>
i4             |dots and/or
jx             |commas have
decimalscale.  |been clipped
|Fetchnum is   |as for trail
|eg the main   |& lead spaces
|called after  |It rounds
|typed-in      |when too many
|numbers with  |decimals, etc

<h5511>
formatnum=     |rightaligned
|In:n1,n2,n3   |so len=14 for
|Gives:quote   |all results;
|Action:n3=num |'decimalsign'
|N1=flag "use  |is fastvar
|groupsigns?"; s3
|N2=decimalqty s2
|n2=0:wholenum s1

<h5512>
i2             |avoid'undef'
i3
makenumber     tx
```

```
f              i2
fixsign        jx
|qtydecimals   |qtydecimals
|tx1=>newtx    |tx1
digitfills     indecimalsign

<h5513>
i1             i2
n?             jx


               |qtydecimals
               |tx1=>newtx
               innumgroup
d4             tx

<h5514>
14             |newlen





               |tx1=>newtx
jx             rightalign.

<h5515>
|To ad or su   |decimalmm &
|decimalnums,  |decimalrd,di;
|use scaleint  |Beware: many
|first, then   |decimals
|ad or su; to  |limit upper
|do mm,rd & di |range of rd &
|the decimal   |di far more
|versions are  |than for mm

<h5516>
decimalmm=     s9
|In:n1,n2,a1   |normalize two
|Gives:num     |numbers to
|As 'n1 n2 mm' |same qty of
|but a1 is     |decimals
|'qty tento':  |to use these
|where qty is  s4
|qty decimals  s3

<h5517>
i3             i3
i9             i9
di             mo
s5             t5
i4             i4
i9             i9
```

```
di             mo
s6             t6

<h5518>
i5             ad
i6             i6
mm             j5
i9             mm
mm
i5
j6
mm             ad

<h5519>
j5             |See note at
j6             |decimalrd
mm             |about whole
i9             |vs decimal
rd             |numbers


ad.

<h5520>
decimalrd=     s9
|In:n1,n2,a1   |note: RANGE,
|Gives:num     |number must
|As 'n1 n2 rd' |be capable of
|but a1 is     |being mm with
|'qty tento':  |a1
|where qty is  s4
|qty decimals  s3

<h5521>
i3             i3
i9             i9
di             mo
s5             t5
i4             i4
i9             i9
di             mo
s6             t6

<h5522>
i9             ad
ni             |Note that
i5             |the formatnum
mm             |&fetchnum are
               |for econ apps
i9             |{unnecessary
j5             |w/technumber
mm             |'undefined'}
```

```
<h5523>
i9             rd.
i6             |Note:g15 pmn
mm             |prefers to
               |order reality
               |by WHOLE
               |numbers; but
j6             |SOME use of
ad             |decimals ok

<h5524>
decimaldi=     s9
|In:n1,n2,a1   |NOTE: range,
|Gives:num     |number must
|As 'n1 n2 di' |be capable of
|but a1 is     |being mm with
|'qty tento':  |a1
|where qty is  s4
|qty decimals  s3

<h5525>
i3             i3
i9             i9
di             mo
s5             t5
i4             i4
i9             i9
di             mo
s6             t6

<h5526>
i9             ad
ni
i5
mm

i9
j5
mm

<h5527>
i9             di.
i6             |when that
mm             |which is
|Decimalrd&di  |divided at is
|are for div   |a whole num,
|by decimnums; |instead use
j6             |ordinary 'rd'
ad             |and 'di'

<h5528>
|The foregoing |or without
|cards contain |'groupsigns'
```

```
|a number of   |{eg comma}
|routines for  |and with or
|elegant       |without
|parsing and   |decimal dot,
|formatting of |cfr fetchnum,
|numbers with  |formatnum,etc

<h5529>
parseyenter=   |all give flag
|In:quote      |dance; while
|Gives:flag    |any other,
|Action:parses |flag basis;
|quote after   |this clips
|eg l9 so that |trail,lead
|y & Y, and a  s3
|pure <enter>, dance

<h5530>
i3             i3
lk             cliptrail
n?             clipleading
|{eg use kb    uppercase
se             up
|before l9}    lk
ex             89
sh             eq.

<h5531>
solvegram=     |Can be used
|In:pos,warp   |instead of wk
|Gives:num     |when basis is
|Actions:reads |to be kw'ed
|content at    |to same place
|pos plus warp |afterwards
|which is then sx
|set to basis  s8

<h5532>
i8             |solvegram: as
ix             |Foundation's
wk             |telegrams
               |{in Asimov's
0              |scifi} which,
i8             |after being
ix             |read, at once
kw.            |dissolves

<h5533>
flipgram=      |then applied
|In:pos,warp   |to the
|Gives:num     |content, so
|Actions:reads |the 'telegram
|content at    |after read is
```

```
|pos plus warp |flipped'
|and "n?"      sx
|operator is   s8

<h5534>
i8
ix
wk
f
n?
i8
ix
kw.

<h5535>
fnextraval=    12
|in:fn#
|gives:number
|Action:picks
|extravalue
|from main
|triplet
fnwarp         wk.

<h5536>
fetchl1main=
|In: fnwarp
|Gives: link1
|main value

50
wk
fnmainval.

<h5537>
l1maintobasis= 0
|In: fnwarp
|Action: link1
|main value,
|pos 10, is    50
|set to basis  jx
               wk
tx             setfnmainval.

<h5538>
l1maintodance= 1
|In: fnwarp
|Action: link1
|main value,
|pos 10, is    50
|set to basis  jx
               wk
tx             setfnmainval.
```

```
<h5539>
link2tr1warp=  51
|In: fnwarp    wk
|Gives: warp   fnwarp
|Action: picks
|link#2, finds
|warp to this,
|and adds 10   10
|to triplet#1  ad.

<h5540>
performfn=     |when called
|In:tr#,fnwarp |fn has its
|Action:to     |active flag
|perform the   |to basis, in
|fnact of a    |contrast to
|triplet of    |'permuteacts'
|any given     tx
|foundry, even f

<h5541>
s8             i1
tripletpos
up             n?
jx
wk             se


s1             ex

<h5542>
i8             fnactlist
jx             lk
|Input to      ay
|fnact
               |And,
               |do it,
|fnact#:       |PerFect:
i1             pf.

<h5543>
doit=          |can also be
|In:fundname   |used as
|Action:calls  |programmers'
|the fnact in  |checking,
|the first     |interactively
|triplet of    |of some types
|fund named    |of funds with
|this;         |fnacts in'em

<h5544>
fnamw
```

```
1

w
performfn.

<h5545>
togglefn=      tx
|In: fn#       49
|Action:       jx
|toggles the   wk
|active-flag   n?
|of a foundry  49
|as for acts   jx
fnwarp         kw.

<h5546>
pffundin=      |fnact in this
|In:value,tr#, |triplet {like
|fnwarp        |performfn};
|Action: Puts  tx
|value n to    f
|extranumber   sx
|of triplet,   tripletpos
|and calls on  f

<h5547>
s3             f
u2             s5
jx
kw             n?

f3             se
jx
wk             ex

<h5548>
ix             fnactlist
               lk
               ay
jx
|Tr#,fnwrp

|Func#
i5             pf.

<h5549>
pffundintwice= s4
|In:firstvalue s3
|secondvalue   s2
|tr# fnwarp    s1
```

```
|Action:exact  |ie, first w/
|like pffundin |firstvalue,
|but does it   |then again
|twice over    |w/secondvalue

<h5550>
i1             i2
i3             i3
i4             i4




pffundin       pffundin.

<h5551>
pffundinscale= |qty5 decimals
|In:qty1,qty5, |to qty1 first
|val,tr#,fnwrp |Then, as a
|Action: as    |second number
|pffundin, but |it gives qty1
|does scaleint |to pffundin;
|of val to     |simplifies
|convert from  |formulacalc

<h5552>
s5             |presumes that
s4             |input has
s3             |already been
t5             |verified
s1             j5
|qty1 is i1    i1
|qty5 is j5    i3
|val is i3     scaleint

<h5553>
i4             i1
i5             i4
               i5




pffundin       pffundin.

<h5554>
pffundgives=   |from same
|In:tr#,fnwarp |triplet &
|Gives:value   |gives it
|Action: Like  tx
|performfn,    f
|does the act, sx
|then picks    tripletpos
```

```
|extravalue    s3

<h5555>
basis          f
|In case       s5
|fnact
|absent        n?

f3             se
jx
wk             ex

<h5556>
sh             fnactlist
               lk
               ay
ix

jx

i5             pf

<h5557>
i3
u2





jx             wk.

<h5558>
pffundingives= |extraval from
|In:n,tr#,fwrp |triplet again
|Gives:value   |& puts to stk
|Action: Puts  tx
|n to extraval f
|of triplet,   sx
|runs this act tripletpos
|then fetches  s3

<h5559>
|In value      i3
|at stk        u2




               jx
               kw

<h5560>
```

```
basis          f
               s5

               n?

f3             se
jx
wk             ex

<h5561>
sh             fnactlist
               lk
               ay
ix

jx

i5             pf

<h5562>
i3
u2





jx             wk.

<h5563>
txquotefnaya=  txquotefnaya
^.



250
sz
&&             kl

<h5564>
quotefnaya=    |a quote out
|In:fn#        |of chars in
|Gives:quote   |an fnaya
|Action: with  |style array;
|len as #1 &   |be sure that
|WITH A NIL    |maxlen is
|AFTER IT,     |ca 232 when
|this makes    |using this

<h5565>
s5             ll:2000000000
txquotefnaya
lk             |This also
```

```
s4              |handles empty
                |arrays
0               i1
f4              i5
kl              fnaya

<h5566>
f               d2
m1
i4              twobillion
ya              s1

                lo

ye              i4.

<h5567>
maxfundnum=     &&
10000.          fundnet
150             kl
maxfundnum
mm              150
200             maxfundnum
ad              fundnet
sz              wwyymatrix

<h5568>
fundnet         |than enough
lk              |of nodes;
thisfcmnet      |next, we use
kl              |4x135 by
|For this       |analogy to
|spreadsheet,   |the 30x50 fcm
|this is        |examples
|much more      |inside tf

<h5569>
maxfundnum      fcmindqty
50              basisthis
ad              maxfundnum
sz              thisfcmnet
                lk
&&              fcmindex
fcmindex        lk
kl              initwarpindex

<h5570>
pos4x135=       |and where
|in:x 1->4;     |four extra
|   y 1->135    |nodes are
|Gives: fn#     |made for
|Suitable       |each, 2 pairs
|when 4x135     |of arrays
```

```
|nodes are     s9
|the 1st ones  sx

<h5571>
ix             ad.
dc
5
mm
i9
dc
20
mm

<h5572>
warp4x135=     ix
|in:x 1->4;    dc
|   y 1->135   5
|Gives: warp   mm
|{'9' looks a  i9
|bit like y:}  dc
s9             20
sx             mm

<h5573>
               ad


               0
               w
               thisfcmnet
               lk
               w9.

<h5574>
mainget4x135=  ix
|in:x 1->4;    dc
|   y 1->135   5
|Gives: main   mm
|value of      i9
|triplet#1     dc
s9             20
sx             mm

<h5575>
               ad


               10
               w
               thisfcmnet
               lk
               ww.
```

```
<h5576>
extraget4x135= ix
|in:x 1->4;    dc
|   y 1->135   5
|Gives: extra  mm
|value of      i9
|triplet#1     dc
s9             20
sx             mm

<h5577>
               ad


               12
               w
               thisfcmnet
               lk
               ww.

<h5578>
mainput4x135=  ix
|in:value,x,y  dc
|Action: puts  5
|in new value  mm
|of tri#1      i9
s9             dc
sx             20
|Stack:value   mm

<h5579>
               ad


               10
               w
               thisfcmnet
               lk
               yy.

<h5580>
fund4x135=     |for each
|in:fn#        |main position
|gives:x,y
|where, in
|contrast to
|fund30x50,    |also,
|we've five    |4x135 funds
|extra funds   |begin at 1

<h5581>
5              w
di
```

```
f
4              4
mo             di
up             up.

<h5582>
columnheader=  64
|In: number    i3
|Action: shows ad
|Column letter jx
|at screentop  up
s3
& &
tx             kl

<h5583>
jx             59
m3
columnspace    60
mm             bt
110            boldbx
ad
f              255
sx             bt.

<h5584>
rownumbercls=  s4
|In: a, b      s3
|Action:this
|clrs prev#, &
|shows row# b  i4
|when startrow makenumber
|#a is on top
|of screen     tx

<h5585>
i4
i3
su
33
mm
87
ad
sx

<h5586>
766            ix
ix             19
               ad
5
               80
```

```
su

807            rt

<h5587>
jx

766

ix


bx.

<h5588>
allrownums=    sx
|In: start#
|Action:
|clears
|earlier
|romnumbers
|and shows
|new ones

<h5589>
ll:21          rownumbercls

ix

ix
m1

ad             lo.

<h5590>
rownumber=     s4
|In: a, b      s3
|Action: Draws
|rowlines &
|shows row# b  i4
|when startrow makenumber
|#a is on top
|of screen     tx

<h5591>
jx             33
               mm
766
               87
i4             ad
i3             f
               sx
su             bx
```

```
<h5592>
14             764

               ix
ix             21
20             ad

               180
ad             rt.

<h5593>
sayctrtactive=
|Action: says
|calculation
|as toggled by
|ctr-t is
|active
|on righthand
|of screen

<h5594>
918            &ACTIVE&
189            920

1000
205

80             195
rt             boldbx.

<h5595>
sayctrtslash=
|Action: says
|calculation
|as toggled by
|ctr-t is
|"//////"
|on righthand
|of screen

<h5596>
918            &//////&
189            920

1000
205

80             195
rt             boldbx.

<h5597>
spreadinfo=    815
```

```
|Action: tells 62
|main keys at
|right part of
|screen and
|tells ctr-t
|is 'active'
&Navigate:&    bx

<h5598>
&PgUp&         &Arrows&
815            815
87             137
bx             bx
&PgDn&
815
112
bx

<h5599>
&CTR-T:&       &calc,&
920            920
100            150
bx             bx
&Toggle&
920
125
bx

<h5600>
&now:&         &______&
920            920
170            204
bx             bx
&______&
920
200
bx             sayctrtactive

<h5601>
&<HOME>:&      &formula&
815            815
180            230
bx             bx
&Enter&        &such as&
815            815
205            255
bx             bx

<h5602>
^a1 b4 add     &or, longer,&
815            815
280            330
bx             bx
```

```
^d3 d2 sub     ^c1:c5 sum
815            815
305            355
bx             bx

<h5603>
^85 mul 5 div  &to put in&
850            815
380            450
bx             bx
&<ENTER>:&     ^text, numbers
815            815
425            475
bx             bx

<h5604>
&<F3>:&        &<F8>:&
815            815
520            590
bx             bx
&Store, etc&   ^Decimals, etc
815            815
545            615
bx             bx

<h5605>
&<INSERT>:&    &add/rm rows&
815            815
660            710
bx             bx
&Repeat char/& &<ESC>: exit&
815            815
685            750
bx             bx.

<h5606>
spreadscreen=  |Comment: look
|Action: shows |at source for
|startup blank |'rt' & we see
|spreadsheet   |that given
|screen with   |a b c d e, it
|ctr-t toggle  |starts at a b
|shown to be   |but completes
|'active';     |before c d

<h5607>
0              1
0              1
1024           1023
768            50
255            0
```

```
rt             rt

<h5608>
765            1
50             51
1023           1
767            767
80             0


rt             rt

<h5609>
2              2
51             766
14             1023
766            766
0              0


rt             rt

<h5610>
15             ll:4
73
765            i1
75             columnheader
0


rt             lo

<h5611>
ll:21          rownumber

1
f
m1             lo


ad             spreadinfo.

<h5612>
positionname=  |input with
|In: x, y      |x as 1 to 4
|Gives: text   |and y as 1 to
|Action: in    |135; fnnames
|own quotespac |have mininum
|gives "vala1" |length 3
|to "vald135"  s6
|given         s5

<h5613>
```

```
^valxnnnnnnnnn
tx             kl
i5             i6
96             makenumber
ad             sx
jx             ix
4              lk
ad             t7

<h5614>
|From:         |Assumption:
ix             |that input
up             |is correct!

|To:           |Qty:
jx             j7
5
ad             fw

<h5615>
j7
4
ad



jx
setlenandnil   jx.

<h5616>
parseposname=  |assumes
|in:quote      |exact length;
|gives:x,y     |{basis x,y
|Action: For   |means that
|lowercase a1  |it didn't
|to d135 gives |parse};
|x y, x=1 to 4
|y=1 to 135;   s4

<h5617>
0              iswithin
0              n?
i4
lk             se
f
s5
2
4              ex

<h5618>
f4             iswithin
               n?
lk
```

```
f              se
s5

97
100            ex

<h5619>
f4             iswithin
up             n?
lk
               se



49
57             ex

<h5620>
f4             iswithin
|Easy convert  n?
|num up to nil
cn             se
f
s7
1
135            ex

<h5621>
sh             i5
               96

               su




sh             i7.

<h5622>
colonrange=    tx
|in:text       |Assumes
|gives:fromfn# |lowercase;
|tofn#         |This gives
|action:4x135  |signed1 twice
|format of     |in case of eg
|ranges such   |outrange or
|as d1:d9      |lack of colon

<h5623>
jx             iswithin
lk
s5             d3
|a1:a1 len 5
|d135:d135 9
i5             oneminus
```

```
5              oneminus
9              ex

<h5624>
&:&            ye
3
jx             d3
3
ad
in             oneminus
s8             oneminus
i8             ex

<h5625>
i8             |From,to,qty:
jx             jx
su             up
s7             ix
               up
&x:xxx&        i7
               dc
sx             fw

<h5626>
i7
dc
ix
setlenandnil
ix
parseposname
s2
s1

<h5627>
|From,to,qty:  i5
i8             i7
up             su
               fw


ix
up

<h5628>
i5             s4
i7             s3
su
ix
setlenandnil

ix
parseposname
```

```
<h5629>
i1             d3
ye

i3
ye             oneminus
               oneminus

an             ex

<h5630>
i1
i2
pos4x135
|Everything
|Is Perfect
i3
i4
pos4x135.

<h5631>
ministkputin=  tx
|in:tr#,fnwarp tripletpos
|new value to  s3
|ministk{365}  |Eg call via
|from extraval |performfn in
|of this tr#;  |a passive fn;
|stkptr:main   |link#1 is to
|value of tr#1 |fnaya datafns

<h5632>
10             |Note that
jx             |when stack
wk             |full, it
s1             |doesn't put
|stkpointer    |in the value
i1             se
365
ge             ex

<h5633>
i3             f1
u2
jx             50
               jx
               wk
               |Array
wk
|New value     fnyay

<h5634>
f1
```

```
10             &ministkputin&
jx             3355
kw.            fnactcherish

<h5635>
ministkget=    tx
|in:tr#,fnwarp tripletpos
|value from    s3
|ministk{365}  |Eg call via
|to extraval   |performfn in
|of this tr#;  |a passive fn;
|stkptr:main   |link#1 is to
|value of tr#1 |fnaya datafns

<h5636>
10             |Note that
jx             |when stack
wk             |empty, it
s1             |signals this
|stkptr is i1  |by
0              |<--basis
i3             jx
u2             kw

<h5637>
i1             m1
1
lt

se
               10
               jx
ex             kw

<h5638>
i1             fnaya

               |value
               |stored;
50             i3
jx             u2
wk             jx
|Array         kw.

<h5639>
```

```
               &ministkget&
               3358
               fnactcherish

<h5640>
ministkreset=  basis
|in:tr#,fnwarp 10
|Action: sets  jx
|stkptr, main  kw.
|value of tr#1
|to basis      &ministkreset&
tx             3340
sh             fnactcherish

<h5641>
tenandextra=   tx
|in:fn#
|gives:n1,n2   37
|action:picks  jx
|mainvalue in  wk
|tr#10 & tr#1  12
|extravalue    jx
fnwarp         wk.

<h5642>
execformbit=   s3
|in:qtydecim,  s2
|type, number, s1
|warp to mstk  s9
|Subroutine of |typenumbers:
|formcalcloop  |1 num to stk
|invokes real  |2 pos to stk
|calculations  |3 do operator

<h5643>
|For type#1,   |fetch pos12 &
|number & qty  |harmonise qty
|decimals each |decim &'push'
|have a type#1 |for type#3,
|so 'pushstk'  |just call the
|in tr#1 of    |fnact in
|ministk;      |triplet1 of
|for type#2,   |the operator

<h5644>
i1             i2
1              dance
eq             i3
n?             pffundin



d5             ex
```

```
<h5645>
i1             i9
2              i2
eq             tenandextra
n?             dance
               i3
               pffundinscale

d7             ex

<h5646>
i1             dance
3              i2
eq             fnwarp
n?             performfn



d5             ex.

<h5647>
formcalcloop=  t5
|in:qtydecim,  sx
|arrayfn#,     s9
|warp to mstk  |i9 qtydecimal
|Subroutine of |for the field
|formcalc;Goes |j5 is warp
|thru compiled |to ministk;
|formula&calcs |ix:fnaya fn#

<h5648>
ll:2000000000  i5
|i5=type:1 num n?
|2 pos, 3 func
i1
ix             se
fnaya
s5
h1             ex

<h5649>
i1             i9
ix             i5
fnaya          i8
s8             j5
               execformbit


|i8 is value   lo.

<h5650>
formcalc=      |refreshfn,
|in:tr#,fnwarp |which tells
|EVERY 4x135   |this to act;
|POSITION in   |l#2tx #3comp
|spreadsheet   |l#4stk  t#1:
|has this as   |type; value
|MAIN fnact;   |t#10 main:
|link#1 is to  |qtydecimals

<h5651>
tx             3
sh             eq
|stk:val1 qty1 n?
|val2 qty2 etc |TYPE3=FORMULA
|{qtydecimals}
10             se
jx
wk             ex

<h5652>
50             fnmainval
jx
|Refreshfn;    n?
|this resets
|ministk,calcs se
|&pops 1 &puts
|it to pos 12
wk             ex

<h5653>
52             53
jx             jx
wk             wk
               fnwarp
|This is the   |This is warp
|fnaya with    |to ministk
|compiled form
s5             t6

<h5654>
|Our "compiled |is fund# to a
|formula" has  |pos eg d135;
|type1,num1,   |#3 is fund#
|type2,num2,,  |to operator
|until type    |eg "add" or
|basis; the    |"sum" {which
|types: #1 is  |pops 'from'&
|number; #2    |'to'from stk}

<h5655>
|Do the        10
|reset of      |triplet#10
|ministk in    |has fnact for
|order that    |reset stk
```

```
|each formula  j6
|can start     |fn# for
|cleanly with  |ministk
|it:           performfn

<h5656>
|calculate:    |Get result:
37             5
jx             j6
wk             pffundgives
|qtydecimals   sh
i5             5
j6             j6
formcalcloop   pffundgives

<h5657>
|Put result    |The way the
|to extravalue |formulas are
|of main tripl |handled, each
|in pos fund:  |operator gets
               |& gives pair
12             |of {result,
jx             |qtydecimals}:
kw.            |SAME qtydecim

<h5658>
&formcalc&     |This will
812            |run compiled
               |form of
               |formula when
               |refresh=yes
               |and when
               |type=3,comp
fnactcherish   |formula

<h5659>
makespreadfns= |so two pairs
|Action: sets  |of following
|up the        |can be used
|spreadsheet   |by fnyay &
|nodes with    |fnaya:storing
|names a1 up   |text; and
|to d135; 5    |compiled form
|for each,     |of formula

<h5660>
100000         4
setfundlevel   135
|Above 4x135   pos4x135
|we have fn#   5
|"refreshfn";  ad
|right next    |'5', because
|we have fn#   |2 pairs array
```

```
|"ministk"     t5

<h5661>
ll:135         positionname
ll:4           fneasyact
|Content here: |comment in 2
basis          |cards ahead>
812            4
basis          |qty links
i2             47
i1             adjustfund

<h5662>
j5             f
50             51
adjustfund     adjustfund
|L#2 & #3 to   |l#4: to the
|data & comp:  |ministack
thisfund       u2
lk             52
up             adjustfund

<h5663>
|tr#1, main:   j5
|type 1 num,   11
|2 txt, 3 form ad
|extra: recent |l#4:ministk,
|value type1&3 |plus 1 plus
|tr#10, main:  |10 for data
|quantity      53
|decimals      adjustfund

<h5664>
ll:4           fundlevel
basis          danceup
&safebank&     |AS FOR 4x135:
fneasy         lo
|link#2 w/data
|fnaya has
|len in pos#1
lo             lo

<h5665>
3500000        |This is
setfundlevel   |linked to
               |as #1 from
               |all 4x135,
               |and:
basis          |next,plus 10,
&refreshfn&    |is linked to
fneasy         |as #4

<h5666>
```

```
3600000        basis
setfundlevel   &ministkarray&
|Next, 10 fns  fneasy
|365 values    ll:9
|stack;fnaya;  basis
|this is after &ministkdata&
|refresh & is  fneasy
|#4 in 4x135   lo.

<h5667>
makespreadfns  |acts:#1put,#5
|Main triplet  |get,#10reset
|value stkptr: 3358
basis          23
3355           adjustfund
basis          3340
&ministk&      38
fneasyact      adjustfund

<h5668>
thisfund       |Qty links:
lk             1
togglefn       47
|usually       adjustfund
|called via    &ministkarray&
|such as       fnam
|performfn;    50
|values 365    adjustfund

<h5669>
databas4x135=  warp4x135
|in:x, y       |l#2:len,text,
|action:sets   |completenil
|basis as      |l#3:type,val,
|first of the  |type,val,
|fnaya arrays  |etc, up until
|at link#2 &   |type nil
|link#3        tx

<h5670>
0              0

1              1

51             52
jx             jx
wk             wk
fnyay          fnyay.

<h5671>
addact=        |Qty decimals
|in:tr#,fnwarp |same and ad
|With ministk  |is good for
```

```
|as link#1,    |all nums
|adds two      |Normally
|topmost &     |called via
|puts result   |such as
|back          |'performfn'

<h5672>
50             5
wk             ix
fnwarp         pffundgives
sx             |decimalqty
               sh
               5
               ix
sh             pffundgives

<h5673>
5              ad
ix             |Ad is superb
pffundgives    |for all num
|decimalqty    |types when
s9             |decimqty same
5              1
ix             ix
pffundgives    pffundin

<h5674>
               i9
               |decimqty
               |always on
               |top of
               |ministk
               1
               ix
               pffundin.

<h5675>
25000000       basis
setfundlevel   328
|Name same as  basis
|in formula:   &add&
               fneasyact
&addact&       thisfund
328            lk
fnactcherish   togglefn

<h5676>
               |Qty links:
               1
               47
               adjustfund
               &ministk&
               fnam
               50
               adjustfund

<h5677>
mulact=        |Qty decimals
|in:tr#,fnwarp |same, in this
|With ministk  |approach,
|as link#1,    |when such
|multiplies    |functions are
|topmost &     |called via
|puts result   |such as
|back          |'performfn'

<h5678>
50             5
wk             ix
fnwarp         pffundgives
sx             |decimalqty
               sh
               5
               ix
sh             pffundgives

<h5679>
5
ix
pffundgives
|decimalqty
s9
5
ix
pffundgives

<h5680>
i9             decimalmm




               1
               ix
tento          pffundin

<h5681>
               i9
               |decimqty
               |always on
               |top of
               |ministk
               1
               ix
               pffundin.
```

```
<h5682>
               basis
               339
|Name same as  basis
|in formula:   &mul&
               fneasyact
&mulact&       thisfund
339            lk
fnactcherish   togglefn

<h5683>
               |Qty links:
               1
               47
               adjustfund
               &ministk&
               fnam
               50
               adjustfund

<h5684>
subact=        |Qty decimals
|in:tr#,fnwarp |same and ad
|With ministk  |is good for
|as link#1,    |all nums
|subs topmost  |Normally
|from 2nd to   |called via
|top & puts    |such as
|result back   |'performfn'

<h5685>
50             5
wk             ix
fnwarp         pffundgives
sx             |decimalqty
               sh
               5
               ix
sh             pffundgives

<h5686>
5              w
ix
pffundgives
|decimalqty
s9
5
ix
pffundgives

<h5687>
               su
               1
               ix
               pffundin

<h5688>
               i9
               |decimqty
               |always on
               |top of
               |ministk
               1
               ix
               pffundin.

<h5689>
               basis
               321
|Name same as  basis
|in formula:   &sub&
               fneasyact
&subact&       thisfund
321            lk
fnactcherish   togglefn

<h5690>
               |Qty links:
               1
               47
               adjustfund
               &ministk&
               fnam
               50
               adjustfund

<h5691>
divact=        |Qty decimals
|in:tr#,fnwarp |same, in this
|With ministk  |approach,
|as link#1,    |when such
|rounddivs 2nd |functions are
|on topmost    |called via
|puts result   |such as
|back          |'performfn'

<h5692>
50             5
wk             ix
fnwarp         pffundgives
sx             |decimalqty
               sh
               5
               ix
sh             pffundgives

<h5693>
5
ix
pffundgives
|decimalqty
s9
5
ix
pffundgives

<h5694>
w              decimalrd




               1
i9             ix
tento          pffundin

<h5695>
               i9
               |decimqty
               |always on
               |top of
               |ministk
               1
               ix
               pffundin.

<h5696>
               basis
               331
|Name same as  basis
|in formula:   &div&
               fneasyact
&divact&       thisfund
331            lk
fnactcherish   togglefn

<h5697>
               |Qty links:
               1
               47
               adjustfund
               &ministk&
               fnam
               50
               adjustfund
```

```
<h5698>
divwact=       |of decimalrd,
|in:tr#,fnwarp |plain rd is
|With ministk  |used; this
|as link#1, as |converts top-
|div, but with |most to whole
|higher range  |in case it
|for decimnum  |isn't already
|for instead   |a whole num

<h5699>
50             5
wk             ix
fnwarp         pffundgives
sx             |decimalqty
               sh
               5
               ix
sh             pffundgives

<h5700>
s5             5
               ix
               pffundgives
               |decimalqty
               s9
               5
               ix
               pffundgives

<h5701>
i9             rd
0
i5


               1
               ix
scaleint       pffundin

<h5702>
               i9
               |decimqty
               |always on
               |top of
               |ministk
               1
               ix
               pffundin.

<h5703>
               basis
               531
|Name same as  basis
|in formula:   &divw&
               fneasyact
&divwact&      thisfund
531            lk
fnactcherish   togglefn

<h5704>
               |Qty links:
               1
               47
               adjustfund
               &ministk&
               fnam
               50
               adjustfund

<h5705>
sumcolumn=     sorttwo
|in:qtydecim,  s4
|x,y1,y2,      s3
|ministk       s2
|              s9
|Subroutine
|of sumact     0
tx             t1

<h5706>
ll:2000000000  s5
               |Desired:
i2             i9
i3             i5
pos4x135       scaleint
|Qtydecimals & j1
|value to stk: ad
tenandextra    t1

<h5707>
h3             j1
i3             i9
i4             dance
le             jx
               pffundintwice
               ex

d6             lo.

<h5708>
sumrow=        sorttwo
|in:qtydecim,  s4
|y,x1,x2,      s3
|ministk       s2
```

```
|              s9
|Subroutine
|of sumact     basis
tx             t7

<h5709>
ll:2000000000  s5
               |Desired:
i3             i9
i2             i5
pos4x135       scaleint
|Qtydecimals & j7
|value to stk: ad
tenandextra    t7

<h5710>
h3             j7
i3             i9
i4             dance
le             jx
               pffundintwice
               ex

d6             lo.

<h5711>
sumact=        |Note, either
|in:tr#,fnwarp |x1=x2 or
|With ministk  |y1=y2; other
|as link#1;    |in increasing
|At ministack: |sequence;
|fromfn, tofn; |2*2 extranums
tx
sh

<h5712>
               ll:4
               5
               ix
50             pffundgives
jx             sh
wk             |'buffer' for
fnwarp         |xn:xn ranges
sx             lo

<h5713>
|i9 is         5
|quantity of   ix
|desired       pffundgives
|decimals in   s9
|the big rich
|sum :)
```

```
<h5714>
|i3 is fn#     5
|from,         ix
|i4 is fn#     pffundgives
|to,           s4
|assuming      5
|equal column# ix
|or equal row# pffundgives
|in sum loop   s3

<h5715>
i3             twobillion
i4             basis
eq             |when no
n?             |arguments
               dance
               ix
               pffundintwice
d6             ex

<h5716>
i3
fund4x135
t2
t1
i4
fund4x135
t4
t3

<h5717>
|same xcoor?   |qtydecimals
|{typically}   i9
j1             j1
j3             j2
eq             j4
n?             ix
               sumcolumn
d7             ex

<h5718>
|same row?     |qtydecimals
               i9
j2             j2
j4             j1
eq             j3
n?             ix
               sumrow
d7             ex

<h5719>
```

```
|In case       basis
|both row      i9
|and column
|numbers
|differ,
|sumact        1
|just gives    ix
|basis         pffundintwice.

<h5720>
|parameters    basis
|has parsing,  395
|but name as   basis
|in formula:   &sum&
               fneasyact
&sumact&       thisfund
395            lk
fnactcherish   togglefn

<h5721>
               |Qty links:
               1
               47
               adjustfund
               &ministk&
               fnam
               50
               adjustfund

<h5722>
avecolumn=     sorttwo
|in:qtydecim,  s4
|x,y1,y2,      s3
|ministk       s2
|              s9
|Subroutine
|of aveact     0
tx             t1

<h5723>
i4
i3
su
up



t9

<h5724>
ll:2000000000  s5
               |Desired:
i2             i9
```

```
i3             i5
pos4x135       scaleint
|Qtydecimals & j1
|value to stk: ad
tenandextra    t1

<h5725>
h3             j9
i3             rd
i4             i9
le             dance
               jx
d8             pffundintwice
               ex
j1             lo.

<h5726>
averow=        sorttwo
|in:qtydecim,  s4
|y,x1,x2,      s3
|ministk       s2
|              s9
|Subroutine
|of aveact     basis
tx             t7

<h5727>
i4
i3
su
up



t9

<h5728>
ll:2000000000  s5
               |Desired:
i3             i9
i2             i5
pos4x135       scaleint
|Qtydecimals & j7
|value to stk: ad
tenandextra    t7

<h5729>
h3             j9
i3             rd
i4             i9
le             dance
               jx
d8             pffundintwice
```

```
               ex
j7             lo.

<h5730>
aveact=        |Note, either
|in:tr#,fnwarp |x1=x2 or
|With ministk  |y1=y2; other
|as link#1;    |in increasing
|At ministack: |sequence;
|fromfn, tofn; |2*2 extranums
tx
sh

<h5731>
               ll:4
               5
               ix
50             pffundgives
jx             sh
wk             |'buffer' for
fnwarp         |xn:xn ranges
sx             lo

<h5732>
|i9 is         5
|quantity of   ix
|desired       pffundgives
|decimals      s9




<h5733>
|i3 is fn#     5
|from,         ix
|i4 is fn#     pffundgives
|to,           s4
|assuming      5
|equal column# ix
|or equal row# pffundgives
|in ave loop   s3

<h5734>
i3             twobillion
i4             basis
eq             |when no
n?             |arguments
               dance
               ix
               pffundintwice
d6             ex
```

```
<h5735>
i3
fund4x135
t2
t1
i4
fund4x135
t4
t3

<h5736>
|same xcoor?   |qtydecimals
|{typically}   i9
j1             j1
j3             j2
eq             j4
n?             ix
               avecolumn
d7             ex

<h5737>
|same row?     |qtydecimals
               i9
j2             j2
j4             j1
eq             j3
n?             ix
               averow
d7             ex

<h5738>
|In case       basis
|both row      i9
|and column
|numbers
|differ,
|aveact        1
|just gives    ix
|basis         pffundintwice.

<h5739>
|parameters    basis
|has parsing,  400
|but name as   basis
|in formula:   &ave&
               fneasyact
&aveact&       thisfund
400            lk
fnactcherish   togglefn

<h5740>
               |Qty links:
               1
               47
               adjustfund
               &ministk&
               fnam
               50
               adjustfund

<h5741>
remact=        |That is,
|in:tr#,fnwarp |plain mo is
|With ministk  |used; this
|as link#1,    |converts top-
|remainder, ie |most to whole
|mo, by        |in case it
|analogy with  |isn't already
|divw          |a whole num

<h5742>
50             5
wk             ix
fnwarp         pffundgives
sx             |decimalqty
               s9
               5
               ix
sh             pffundgives

<h5743>
s5             5
i9             ix
0              pffundgives
i5             |decimalqty
scaleint       sh
               5
               ix
s5             pffundgives

<h5744>
s3             0
i9             i9
0              i6
i3             scaleint
scaleint
i5             1
mo             ix
s6             pffundin

<h5745>
               i9
               |decimqty
               |always on
               |top of
               |ministk
               1
               ix
               pffundin.

<h5746>
               basis
               431
|Name same as  basis
|in formula:   &rem&
               fneasyact
&remact&       thisfund
431            lk
fnactcherish   togglefn

<h5747>
               |Qty links:
               1
               47
               adjustfund
               &ministk&
               fnam
               50
               adjustfund

<h5748>
permilleact=
|in:tr#,fnwarp
|With ministk
|as link#1,
|permille
|operator
|in the
|spreadsheet

<h5749>
50             5
wk             ix
fnwarp         pffundgives
sx             |decimalqty
               sh
               5
               ix
sh             pffundgives

<h5750>
5
ix
pffundgives
|decimalqty
s9
5
ix
pffundgives
```

```
<h5751>
i9             decimalmm
               1000
               rd


               1
               ix
tento          pffundin

<h5752>
               i9
               |decimqty
               |always on
               |top of
               |ministk
               1
               ix
               pffundin.

<h5753>
               basis
               458
|Name same as  basis
|in formula:   &permille&
               fneasyact
&permilleact&  thisfund
458            lk
fnactcherish   togglefn

<h5754>
               |Qty links:
               1
               47
               adjustfund
               &ministk&
               fnam
               50
               adjustfund

<h5755>
ndivact=       |Qty decimals
|in:tr#,fnwarp |same, in this
|With ministk  |approach,
|as link#1,    |when such
|integerdivide |functions are
|2nd on topmst |called via
|&puts result  |such as
|back          |'performfn'

<h5756>
50             5
```

```
wk             ix
fnwarp         pffundgives
sx             |decimalqty
               sh
               5
               ix
sh             pffundgives

<h5757>
5
ix
pffundgives
|decimalqty
s9
5
ix
pffundgives

<h5758>
w              decimaldi




               1
i9             ix
tento          pffundin

<h5759>
               i9
               |decimqty
               |always on
               |top of
               |ministk
               1
               ix
               pffundin.

<h5760>
               basis
               422
|Name same as  basis
|in formula:   &ndiv&
               fneasyact
&ndivact&      thisfund
422            lk
fnactcherish   togglefn

<h5761>
               |Qty links:
               1
               47
               adjustfund
               &ministk&
               fnam
               50
               adjustfund

<h5762>
ndivwact=      |of decimalrd,
|in:tr#,fnwarp |plain di is
|With ministk  |used; this
|as link#1, as |converts top-
|div, but with |most to whole
|higher range  |in case it
|for decimnum  |isn't already
|for instead   |a whole num

<h5763>
50             5
wk             ix
fnwarp         pffundgives
sx             |decimalqty
               sh
               5
               ix
sh             pffundgives

<h5764>
s5             5
               ix
               pffundgives
               |decimalqty
               s9
               5
               ix
               pffundgives

<h5765>
i9             di
0
i5



               1
               ix
scaleint       pffundin

<h5766>
               i9
               |decimqty
               |always on
               |top of
               |ministk
               1
               ix
               pffundin.

<h5767>
               basis
               532
|Name same as  basis
|in formula:   &ndivw&
               fneasyact
&ndivwact&     thisfund
532            lk
fnactcherish   togglefn

<h5768>
               |Qty links:
               1
               47
               adjustfund
               &ministk&
               fnam
               50
               adjustfund

<h5769>
rffgact=       |from one to
|in:tr#,fnwarp |the number on
|With ministk  |ministk; this
|as link#1,    |converts top-
|makes a new   |most to whole
|relatively    |in case it
|free fluct    |isn't already
|number range  |a whole num

<h5770>
50             5
wk             ix
fnwarp         pffundgives
sx             |decimalqty
               t3
               5
               ix
sh             pffundgives

<h5771>
s5             rffg
j3             |a relatively
0              |Free
i5             |fluctuation
               |number
               s9
               0
scaleint       j3

<h5772>
```

```
i9             j3
scaleint       |decimqty
               |always on
               |top of
               |ministk
1              1
ix             ix
pffundin       pffundin.

<h5773>
               basis
               533
|Name same as  basis
|in formula:   &rffg&
               fneasyact
&rffgact&      thisfund
533            lk
fnactcherish   togglefn

<h5774>
               |Qty links:
               1
               47
               adjustfund
               &ministk&
               fnam
               50
               adjustfund

<h5775>
procact=
|in:tr#,fnwarp
|With ministk
|as link#1,
|procent




<h5776>
50             5
wk             ix
fnwarp         pffundgives
sx             |decimalqty
               sh
               5
               ix
sh             pffundgives

<h5777>
5
ix
pffundgives
```

```
|decimalqty
s9
5
ix
pffundgives

<h5778>
i9             decimalmm
               100
               rd


               1
               ix
tento          pffundin

<h5779>
               i9
               |decimqty
               |always on
               |top of
               |ministk
               1
               ix
               pffundin.

<h5780>
               basis
               438
|Name same as  basis
|in formula:   &proc&
               fneasyact
&procact&      thisfund
438            lk
fnactcherish   togglefn

<h5781>
               |Qty links:
               1
               47
               adjustfund
               &ministk&
               fnam
               50
               adjustfund

<h5782>
roundact=
|in:tr#,fnwarp
|With ministk
|as link#1,
|rounds n1 by
|n2 digits as
```

```
|spreadsheet
|function

<h5783>
50             5
wk             ix
fnwarp         pffundgives
sx             |decimalqty
               t3
               5
               ix
sh             pffundgives

<h5784>
s5
j3
0
i5


scaleint
s8

<h5785>
5              s6
ix             j3
pffundgives    0
|decimalqty    i6
sh
5              scaleint
ix
pffundgives    s6

<h5786>
i8             i6
0
9              i9
makefit        rd
tento

               i9
s9             mm

<h5787>
s5             j3
0              |decimqty
j3             |always on
i5             |top of
scaleint       |ministk
1              1
ix             ix
pffundin       pffundin.
```

```
<h5788>
               basis
               550
|Name same as  basis
|in formula:   &round&
               fneasyact
&roundact&     thisfund
550            lk
fnactcherish   togglefn

<h5789>
               |Qty links:
               1
               47
               adjustfund
               &ministk&
               fnam
               50
               adjustfund

<h5790>
cleansetopbox= 3
|Action:       3
|clears the    1021
|top box of    49
|the screen,
|called eg
|after         0
|input         rt.

<h5791>
eleganttxt01=  cliptrail
^.

longtxt*
***** Complete
d spreadsheet
run ********   eleganttxt01
*txtcomplete   kl

<h5792>
elegantleave=  fnloopcont
|Called by
|spreadsheet
|when <esc>
|is clicked


cleansetopbox  basisthis

<h5793>
eleganttxt01   400
lk             330
```

```
               af
               ad

120
25
bx             activepause.

<h5794>
posclswhite=   |in: x, y
|Action:       |{range: x is
|normalises    |1 to 4, y is
|position in   |1 to 21, ie,
|spreadsheet   |visible part
|after marker  |of spread}
|or to update  s4
|value         s3

<h5795>
m3             mm
columnspace    76
mm
33
ad

m4
33             ad

<h5796>
d              255
columnspace
ad
up

d
30
ad             rt.

<h5797>
posclsblack=   |in: x, y
|Action: shows |{range: x is
|marker of     |1 to 4, y is
|position in   |1 to 21, ie,
|spreadsheet   |visible part
|by inverted   |of spread}
|tones in the  s4
|position      s3

<h5798>
m3             mm
columnspace    76
mm
33
ad
```

```
m4
33             ad

<h5799>
d              0
columnspace
ad
up

d
30
ad             rt.

<h5800>
poswhitetxt=   tx
|In:txt,x,y    jx
|Action:shows  lk
|quote in      n?
|area after eg
|posclsblack   se
s4
s3             ex

<h5801>
jx             m4
               33
               mm
m3             86
columnspace
mm             ad
31
ad             bx.

<h5802>
posblacktxt=   tx
|In:txt,x,y    jx
|Action:shows  lk
|quote in      n?
|area after eg
|posclswhite   se
s4
s3             ex

<h5803>
0              m4
bt             33
jx             mm
m3             86
columnspace    ad
mm             boldbx
31             255
ad             bt.
```

```
<h5804>
stdquotepos=   |Standard in
|in: fnx,fny,  |the sense,
|fnwarp to     |natural form
|fcminterface  |eg formatnum
|qives: quote
|Appropriate   sx
|for view in   |Note 4x135:
|spreadsheet   warp4x135

<h5805>
tx             &&
               i5
10             n?
jx
wk             se

|Type:         ex
s5             sh

<h5806>
i5             jx
2              link2tr1warp
eq             ex
n?
|texts are     |Then type is
|quotes within |number or
|single fn     |formula;gives
d3             |result next:

<h5807>
|#9 "," flag:  12
16             jx
ix             wk
wk
|#10 qtydecim: |numgroupflag
37             |qtydecimals
jx             |the number
wk             formatnum.

<h5808>
xyto4x135=     |The x and y
|in:screenx,   |refers to
|screeny,fnwrp |screen only;
|Gives:fnx,fny |from 1,1 to
|spreadsheet's |4, 21
|fnwarp helps, tx
|convert x,y   s4
|to pos4x135   | x on stack

<h5809>
37
```

```
jx

wk

m4
ad.

<h5810>
maybeshowform= xyto4x135
|in:scrx,scry,
|fnwarp to
|interface
|When node is
|formula, show
|inside topbox warp4x135
|in text form  tx

<h5811>
               10
               jx
               wk


               |is it formula
               3
               eq

<h5812>
n?             &===&

se

ex
               20
|It's a        23
|formula:      bx

<h5813>
               51
               jx
               wk

               quotefnaya
               55
               23
               bx.

<h5814>
maybeclsbox=   xyto4x135
|in:scrx,scry, warp4x135
|fnwarp to     10
|interface     wk
```

```
|When node is
|formula, show
|inside topbox |Formula-type:
|in text form  3

<h5815>
eq

se




cleansetopbox.

<h5816>
refreshxywhi=  |screenfield
|in:screenx,   |as given,
|screeny,      |x and y up to
|fnwarp to     |4x21;
|interface     |white
|Action:shows  tx
|spreadsheet   s5
|content at    s3

<h5817>
i3             stdquotepos
i5
jx


               i3
xyto4x135      i5
jx             poswhitetxt.

<h5818>
whiterefresh3= |but only
|in:screenx,   |does it when
|screeny,      |formula field
|fnwarp to     |ie type 3
|interface
|Action: both  tx
|posclswhite & s5
|refreshxybla  s3

<h5819>
i3             ix
i5             i9
jx             mainget4x135


xyto4x135
s9
```

```
sx              |Type at stk

<h5820>
3               i3
eq              i5
n?

se


ex              posclswhite

<h5821>
                ix
                i9
                jx
                stdquotepos

                i3
                i5
                posblacktxt.

<h5822>
whiterewrite=  |for all
|in:screenx,   |positions
|screeny,
|fnwarp to
|interface
|Action: both  tx
|posclswhite & s5
|refreshxybla  s3

<h5823>
i3
i5
jx


xyto4x135
s9
sx

<h5824>
                i3
                i5





                posclswhite

<h5825>
```

```
               ix
               i9
               jx
               stdquotepos

               i3
               i5
               posblacktxt.

<h5826>
refreshxybla=  |screenfield
|in:screenx,   |as given,
|screeny,      |x and y up to
|fnwarp to     |4x21;
|interface     |black
|Action:shows  tx
|spreadsheet   s5
|content at    s3

<h5827>
i3             stdquotepos
i5
jx


               i3
xyto4x135      i5
jx             posblacktxt.

<h5828>
rewriteallpos= tx
|in:fnwarp to
|interface
|Action: this
|refreshes
|even text &
|value fields
|at screen

<h5829>
|This concerns 10
|all fields    jx
|other than    wk
|present pos,  sx
|which should  12
|be handled    jx
|via the       wk
|marking       s9

<h5830>
ll:21          i2
               ix
               eq
               i1
               i9
ll:4           eq

<h5831>
an             i2
               i1
               jx
               whiterewrite


               lo
d4             lo.

<h5832>
scrollbyarrow= 1
|in:fnwarp     eq
|to interface
tx             se
dance
12             |Find
jx             |direction:
wk             ts

<h5833>
sx             |Arrowscroll
               |can often
1              |lead to
115            |overrepeat,
ix             |so attend to
37             |keyboard-
jx             |buffer:
pn             kb

<h5834>
37             cleansetopbox
jx
wk



               jx
allrownums     rewriteallpos.

<h5835>
markerupdate=  |The x and y
|in:newx,      |refers to
|newy,fnwarp   |screen only;
|to interface  |from 1,1 to
|Subroutine of |4, 21
|fcminterface, tx
```

```
|updates pos   s4
|w/rangefix    s3

<h5836>
10             i5
jx             i6
wk             jx
s5             extratriplet
12
jx
wk
s6             maybeclsbox

<h5837>
               i5
               i6
               posclswhite



               |Extra triplet
               refreshxybla

<h5838>
i3             s2
1              s1
4
makefit
i4
1              i2
21             i4
makefit        eq

<h5839>
d2             i1




               10
jx             jx
scrollbyarrow  kw

<h5840>
i2             i1
12             i2
jx             jx
kw             extratriplet
               refreshxywhi
i1
i2
posclsblack    maybeshowform.
```

```
<h5841>
refreshallpos= tx
|in:fnwarp to
|interface
|Action: this
|updates the
|screeninfo as
|to most
|recent values

<h5842>
|This concerns 10
|the numbers   jx
|of type 3 &   wk
|other than    sx
|present pos,  12
|which is      jx
|always        wk
|updated       s9

<h5843>
ll:21          i2
               ix
               eq


               i1
               i9
ll:4           eq

<h5844>
an             i2
               i1
               jx
               whiterefresh3


               lo
d4             lo.

<h5845>
typedinput=    typedinput
^.



200
sz
&&             kl

<h5846>
storetypednum= t5
|in:txt,fnw1,  tx
```

```
|fnw2           |j5, the field
|Txt has num;   |Will call
|fnwarp1 to     |fetchnum from
|fcminterface;  |quoted num
|fnwarp2 to fn  |in ix
|in pos4x135    sx

<h5847>
|GroupsignTr#3  ix
16              |Groupsigns
jx              |are a global
wk              |setting
|QtydecimTr#10  fetchnum
37              12
j5              j5
wk              kw.

<h5848>
keyenter=       |The x and y
|in:x,y,fnwarp  |refers to
|to interface;  |screen only
|Subroutine of  cleansetopbox
|fcminterface,
|processes      tx
|press of       s4
|<enter> key    s3

<h5849>
|New datafield  wk
|means, when    n?
|ctr-t is at
|active, calc   d2
|new formula
|results
31              jx
jx              l1maintodance

<h5850>
^TYPE CONTENT:  37
20              jx
23              wk
bx              m4
                ad
typedinput
lk              |pos4x135 y:
sx              s5

<h5851>
i3              200
i5              23
warp4x135       14
t7              ix
```

```
i3
i4
posclsblack    l9

<h5852>
basis          cleansetopbox
10
j7
kw
basis
12             i3
j7             i5
kw             databas4x135

<h5853>
               ix
               lk
               n?

               se


               ex

<h5854>
ix             f1
leisurenum



               10
n?             j7
s1             kw

<h5855>
i1             ix




               jx
               j7
d4             storetypednum

<h5856>
51             ix
j7             lk
wk
               u2
               |One extra for
               |len; and one
|Data array    |extra for nil
s9             t1
```

```
<h5857>
ll:2000000000  i1
               |When two fns
               |are used as
               |fnyay array,
               |74 positions
m1             |are available
ix             i9
ay             fnyay

<h5858>
i1             lo
j1
lt

d2

twobillion
s1

<h5859>
               i3
               i5
               jx
               stdquotepos
               i3
|After         i4
|important     poswhitetxt
|entry,good to kb.

<h5860>
compthistx1=   &&
^.
|The formulas  |Each part of
|always within |the formula
|80 chars, so  |gets copied
|ample room:   |over here
250            compthistx1
sz             kl

<h5861>
comppartnum=   tx
|in:num,fn#aya sx
|fnwrp         s5
|action:this
|subroutine of 34
|compthispart  jx
|compiles in a wk
|number        s3

<h5862>
|Compile in    |Compile in
```

```
|type:         |the value:
1              i5
               f3

i3
ix             ix
fnyay          fnyay

<h5863>
|The compile   2
|next
|position in
|triplet#9
|of the        34
|fcminterface  jx
|foundry is    ad
|updated:      ku.

<h5864>
comppartrange= tx
|in:a,b,fn#aya sx
|fnwrp         s2
|action:this   s1
|subroutine of |as numbers:
|compthispart  |qtydecim,fn#1
|compiles in a |fn#2, then
|range         |2*2 nils

<h5865>
|The idea of   i1
|'buffer':     ix
|A func like   jx
|"add" would   comppartnum
|yield basis   i2
|when given a  ix
|range "sum"   jx
|should have   comppartnum

<h5866>
|Fetch qty     ix
|decimals
36
jx
wk
|is in fnwarp
|fcminterface  jx
|as tr#9 vars  comppartnum

<h5867>
|The addition  ll:4
|of two pair   0
|basis'es      ix
|after a range jx
```

```
|is expected   comppartnum
|by eg "ave"
|as a sort of
|'buffer'      lo.

<h5868>
comppartpos=   tx
|in:pos,fn#aya sx
|fnwrp         s5
|action:this
|subroutine of 34
|compthispart  jx
|compiles in a wk
|position      s3

<h5869>
|Compile in    |Compile in
|type:         |the position:
2              i5
               f3

i3
ix             ix
fnyay          fnyay

<h5870>
|The compile   2
|next
|position in
|triplet#9
|of the        34
|fcminterface  jx
|foundry is    ad
|updated:      ku.

<h5871>
comppartfunc=  tx
|in:num,fn#aya sx
|fnwrp         s5
|action:this
|subroutine of 34
|compthispart  jx
|compiles in a wk
|function      s3

<h5872>
|Compile in    |Compile in
|type:         |the function:
3              i5
               f3

i3
ix             ix
```

```
fnyay          fnyay

<h5873>
|The compile   2
|next
|position in
|triplet#9
|of the        34
|fcminterface  jx
|foundry is    ad
|updated:      ku.

<h5874>
compthispart=  |len is the
|in:start,len, |length of
|fn#,fnwrp     |this nonspace
|gives: done?, |unit {fnwrp:
|ok? {on top}  |fcminterface:
|Action:start  |tr#9 has info
|is WARP into  |} fn#:fnaya
|formulaquote, |to store   >>

<h5875>
|okflag is 1,  tx
|otherwise     sx
|it is a       |fnwrp's tr#9:
|syntax issue; |Main:next pos
|doneflag is   |to compile to
|1 when basis  |Xtra:qtydecim
|start {from   s2
|wordcountx}   s1

<h5876>
36             dance
jx             dance
wk             ex
t3
i1             16
ye             jx
               wk
d3             t5

<h5877>
compthistx1    |From,to,qty:
lk             i1
t1
               j1
               up
i2
j1             i2
setlenandnil   fw

<h5878>
```

```
basis
|stk:doneflag
j1
leisurenum

n?

dh

<h5879>
j5             ix
j3             jx
j1             comppartnum
fetchnum       tn
ix             dance
jx             tn
comppartnum    tn
j3             ex

<h5880>
j1             i5
parseposname   n?
s6

               i6
               n?

s5             or

<h5881>
               i5
               i6
               pos4x135
               ix
               jx
               comppartpos
               dance
d8             ex

<h5882>
j1             i6
fnam           ix
s6             jx
i6             comppartfunc
!1             tn
eq             dance
               tn
d8             ex

<h5883>
j1             i5
colonrange     !1
s6             eq
```

```
               i6
               !1
               eq

s5             or

<h5884>
               i5
               i6
               ix
               jx
               comppartrange
               dance
               tn
d8             ex

<h5885>
|textpart      |Doneflag is
|was checked,  |on stk;
|1] is numlike |Stricter chk
|{also range}; |of syntax to
|2] is it      |prefix eg add
|positionlike; |to 'funcadd',
|3] is it      |internal fnam
|operatorname  basis.

<h5886>
tellsyntax=    tx
|in:warppos,tx s5
|action: msg
|when formula  |{quote will
|has just been |be changed
|typed in w/?  |during
|at position   |message}
|warped to     cleansetopbox

<h5887>
&please&       &retry:-)&
|Rule of thumb |pc must show
|as for mind-  |humility,esp
|respecting    |as for making
|programs:     |a request!
4              12
8              29
rp             bx

<h5888>
jx             jx
               lk
               jx
               ns
               |Ascii:'?'
115            63
```

```
29             i5
bx             kl

<h5889>
jx             1200
115            activepause
11             |nuke rolling
bx             |keys away:
               kb
0              ki
jx             sh
setlenandnil   cleansetopbox.

<h5890>
compileform=   |triplet#1main
|in:txt,x,y    |gets 3 at ok
|interfacewrp; |parse {other-
|Compiles txt  |wise 0}:type
|to fnaya in   tx
|l#3; msg at   warp4x135
|issue, then   t1
|txt erased    sx

<h5891>
52             |where type
j1             |1 is number
wk             |2 is fn# pos
|j3 has        |3 is fn# func
|array:        |and formula
|type,value,   |is complete
|type,value,,  |when type is
t3             |basis

<h5892>
basis          basis
10             1
j1             j3
kw             fnyay
basis          basis
12             2
j1             j3
kw             fnyay

<h5893>
1              i4
34             n?
jx
kw             se
|tr#9:compvars
ix             ex
wordcount      |tr#9main:pos#
s4             |in comparray
```

```
<h5894>
|So it has     basis
|content; the  basis
|wordcountx
|may have data i4
|from earlier  up
|in session &  wordcountx
|basis is sign lk
|used next     storepair

<h5895>
ll:2000000000  |Note that
i1             |'wordcount'
wordcountx     |produces
lk             |position as
getpair        |warps into
               |the text
t9             |as a direct
t8             |address

<h5896>
|Input next:   |the
|textposwarp,  |compthispart
|length,       |updates tr#9
|fn# to array, |main value in
|formulatext   |fcminterface
               j3
j8             jx
j9             compthispart

<h5897>
|Okflag        sh
ye             j8
|when ok,      ix
|already       tellsyntax
|compiled in,  |proper msg,
|then skip     |ready to
|next column:  |exit formula
d5             ex

<h5898>
|Doneflag      twobillion
n?             s1
|Anything      |Using2billion
|more in the   |in loop is
|formula?      |in 1sthand
|if so, skip   |knowing of
|next two      |upper limits
d2             lo

<h5899>
|Completion    |This basis
|signal:       |in the
```

```
basis          |compiled
34             |array is
jx             |inserted at
wk             |what tr#9 say
j3             |is 'next
fnyay          |comp pos'

<h5900>
               |All right!
               |Syntax ok as
               |far as it
               |goes:
               3
               10
               j1
               kw.

<h5901>
keyhome=       |are:1=number;
|in:x,y,fnwarp |2=fundnum to
|to interface  |anyone 4x135;
|Subroutine of |3=func eg add
|fcminterface, cleansetopbox
|does <home>;  tx
|Compiled      s4
|formula codes s3

<h5902>
^FORMULA:      37
20             jx
23             wk
cleansetopbox  m4
bx             ad
typedinput
lk             |pos4x135 y:
sx             s5

<h5903>
i3             23
i5             62
warp4x135      ix
t7             l9
i3             ix
i4             cliptrail
posclsblack    lowercase
129            sh

<h5904>
basis          cleansetopbox
10             |After loop,
j7             |set xtraval
kw             |tr#9 in jx to
basis          |qty decimals
```

```
12             i3
j7             i5
kw             databas4x135

<h5905>
ix             basis
lk             |Initial value
               |before
n?             |formula
               |performed
se             12
               j7
ex             kw

<h5906>
51             ix
j7             lk
wk
|Data array    u2
s9             |One extra for
ix             |len; and one
clipleading    |extra for nil
sx             t1

<h5907>
ll:2000000000  i1
               |When two fns
               |are used as
               |fnyay array,
               |74 positions
m1             |are available
ix             i9
ay             fnyay

<h5908>
i1             lo
j1             |qtydecim>tr#9
lt             37
               j7
d2             wk
               36
twobillion     jx
s1             kw

<h5909>
ix             |The
|jx tr#9, both |indication
|values,in use |that the
|in compform   |parsing of
i3             |formula
i5             |showed good
jx             |syntax is
compileform    |that type=3:
```

```
<h5910>
10             i3
j7             i5
wk             |Keep dataarea
               |clean; makes
3              |easier save
eq             |routines

d3             databas4x135

<h5911>
               31
               jx
               wk
               n?
               d2

               jx
               l1maintodance

<h5912>
i3             jx
i4             l1maintodance
jx             |This works
               |no matter
               |parse:
               1
               j7
maybeshowform  performfn

<h5913>
               31
               jx
               wk

               d2

               jx
               l1maintobasis

<h5914>
               i3
               i5
               jx
               stdquotepos
               i3
|After         i4
|important     poswhitetxt
|entry,good to kb.

<h5915>
keypgup=       tx
```

```
|in:fnwarp     cleansetopbox
|to interface  1
|Subroutine of 115
|fcminterface, !21
|processes     37
|press of      jx
|pgup          pn

<h5916>
37
jx
wk




allrownums

<h5917>
10             jx
jx
wk
12
jx
wk
jx
markerupdate   rewriteallpos.

<h5918>
keypgdn=       tx
|in:fnwarp     cleansetopbox
|to interface  1
|Subroutine of 115
|fcminterface, 21
|processes     37
|press of      jx
|pgdn          pn

<h5919>
37
jx
wk




allrownums

<h5920>
10             jx
jx
wk
12
```

```
jx
wk
jx
markerupdate   rewriteallpos.

<h5921>
putcharshere=  tx
|In:char,x,y,
|fnwarp to     jx
|fcminterface  xyto4x135
|Action: puts  s4
|14 chars to   s3
|this field,
|and type=text s7

<h5922>
i3             i7
i4             32
               eq
|Blanks are
|treated as    se
|'erase'

databas4x135   ex

<h5923>
               i3
               i4
               warp4x135




               tx

<h5924>
|Type=text:    |Link#2=data
2              51
10             jx
jx             wk



kw             s5

<h5925>
ll:14          basis
i7             16
|First is p#2  i5
f1             fnyay
i5             14
fnyay          1
               i5
```

```
lo              fnyay.

<h5926>
charrighttx=    nd <ENTER>}===
^.              ==>
                *txtcomplete
longtxt*        cliptrail
Pls type chara
cter to repeat
 from here & t charrighttx
o the right {a kl

<h5927>
charright=      tx
|In:x,y,fnwarp s4
|fcminterface  s3
|Repeats char
|in this
|field, and to typedinput
|any fields to lk
|the right     sx

<h5928>
                charrighttx



                lk
                10
                24
                bx

<h5929>
955             sx
23
1
ix
l9
ix
|Accept space,
|too!           cleansetopbox

<h5930>
ix              ix
lk              up
n?              lk

se


ex              s9

<h5931>
```

```
               i9
               m1

               i3
               ad
               i4
               jx
ll:4           putcharshere

<h5932>
i1             ex
i3
ad
4
gt


se             lo.

<h5933>
blankrowhere=  s5
|in: row#
|action:erases
|column 1->4
|in the 1->135
|rownum; do be
|sure input is
|fully right

<h5934>
ll:4           0
i1             10
i5             jx
databas4x135   kw
i1             0
i5             12
warp4x135      jx
tx             kw

<h5935>
0
37
jx
kw



lo.

<h5936>
copyfieldar1=  tx
|in:fwp1,fwp2  sx
|Ie,fnwarps    |Note that
```

```
|4x135; copies |link#2 fnaya
|f1->f2 text   |has textlen
|array,ie,l#2; |in pos#1,here
|Be sure input |used,and it's
|&arrays right |nilcomplete

<h5937>
51             1
ix             i5
wk             fnaya
s5             s4
51             i4
jx             1
wk             i6
s6             fnyay

<h5938>
ll:2000000000  f1
|Length up up: f4
f1             gt
i5
fnaya          se
f1
i6             ex
fnyay          lo.

<h5939>
copyfieldar2=  tx
|in:fwp1,fwp2  sx
|Ie,fnwarps    |l#3 is fnaya
|4x135; copies |comp formula
|f1->f2 comp   |with type,val
|array,ie,l#3; |type,val,etc,
|Be sure input |from pos#1,to
|&arrays right |type nil

<h5940>
52
ix
wk
s5
52
jx
wk
s6

<h5941>
ll:2000000000  n?
i1
i5             se
fnaya
f              ex
i1
```

```
i6
fnyay          h1

<h5942>
|Pairwise      |The value
i1             |is after
i5             |the type
fnaya          |and can
               |be anything
i1
i6
fnyay          lo.

<h5943>
copyfieldatob= t4
|in: x1, y1,
|x2, y2        t3
|Copies all of |These are
|field x1y1 to |1,1 to 4,135
|x2y2; do be   t2
|sure input is
|fully right   t1

<h5944>
j1             |Data in use
j2             |is copied
warp4x135      |over;
sx             |the
j3             |databas4x135
j4             |is first done
warp4x135      |for
tx             |destination

<h5945>
j3             37
j4             ix
               wk


               37
               jx
databas4x135   kw

<h5946>
10             12
ix             ix
wk             wk
f              |i8=type 1num
s8             |2txt 3formula
10             12
jx             jx
kw             kw
```

```
<h5947>
i8             ix
1              jx
eq

se


ex             copyfieldar1

<h5948>
i8             ix
2              jx
eq

se


ex             copyfieldar2.

<h5949>
copyrowatob=   s8
|in: n1, n2
|action:copies
|row# n1 to    s6
|row# n2, all
|data; do be
|sure input is
|fully right

<h5950>
ll:4
i1
i6
i1
i8
copyfieldatob

lo.

<h5951>
rmrowheretx=   e? Y/N <ENTER>
^.             =Y
               *txtcomplete
longtxt*       cliptrail
Removing a row
 may require f
ormula retype. rmrowheretx
 Sure to remov kl

<h5952>
rmrowhere=     xyto4x135
|In:x,y,fnwarp s4
```

```
|fcminterface  sh
|Removes this
|row, after
|asking        typedinput
|for y/n       lk
|confirm       sx

<h5953>
               rmrowheretx



               lk
               10
               24
               bx

<h5954>
955            parseyenter
23             n?
3
ix             se
kb
l9             ex
cleansetopbox  135
ix             t7

<h5955>
ll:2000000000  d3
m1
i4             j7
ad             blankrowhere
s3
f3
j7
le             ex

<h5956>
f3
i3
copyrowatob




lo.

<h5957>
insertrowtx=   e now? Y/N <EN
^.             TER>=Y
               *txtcomplete
longtxt*       cliptrail
Creating a bla
```

```
nk row {may re
quire formula  insertrowtx
retype}. Creat kl

<h5958>
insertrow=     xyto4x135
|In:x,y,fnwarp s4
|fcminterface  sh
|Inserts new
|row, after
|asking        typedinput
|for y/n       lk
|confirm       sx

<h5959>
               insertrowtx



               lk
               10
               24
               bx

<h5960>
955            parseyenter
23             n?
3
ix             se
kb
l9             ex
cleansetopbox  135
ix             s3

<h5961>
ll:2000000000  d3

i3             i4
i4             blankrowhere
eq


n?             ex

<h5962>
m3
i3
copyrowatob



q3
lo.
```

```
<h5963>
insertmenutx1=  row <3>INSERT
^.              new row here
               *txtcomplete
longtxt*       cliptrail
MENU:<1>REPEAT
 one character
 to the right  insertmenutx1
<2>REMOVE this kl

<h5964>
keyinsert=     tx
|in:x,y,fnwarp
|to interface
|Subroutine of s8
|fcminterface,
|processes
|press of
|<insert> key  s9

<h5965>
cleansetopbox  insertmenutx1
               lk



               10
               24
               bx

<h5966>
ki
48
su
sx



cleansetopbox

<h5967>
ix             i9
1              i8
eq             jx
n?             charright


               jx
d6             rewriteallpos

<h5968>
ix             i9
```

```
2              i8
eq             jx
n?             rmrowhere


               jx
d6             rewriteallpos

<h5969>
ix             i9
3              i8
eq             jx
n?             insertrow


               jx
d6             rewriteallpos

<h5970>
i9
i8




jx
markerupdate.

<h5971>
f3successtx=                {
^.             press SPACE}
               *txtcomplete
longtxt*       cliptrail
The <F3> actio
n completed, i
nvolving this  f3successtx
qty of cards:  kl

<h5972>
f3quote1=      &&
^.
|Can be used
|during
|f3 storage

100            f3quote1
sz             kl

<h5973>
f3cardidisstx= s typed right
^.              PRESS SPACE
               *txtcomplete
longtxt*       cliptrail
```

```
????! Do pls d
o it anew but
this time be s f3cardidisstx
ure location i kl

<h5974>
tellf3success= s5
|In:qty
|Action:tells  f3successtx
|that f3 did   lk
|it good with
|qty cards     10
|involved, &   24
|receives key  bx

<h5975>
i5             kb
makenumber

               ki
               sh
680
24
bx             cleansetopbox.

<h5976>
tellf3cidiss=
|Action: tell
|that there
|was an issue
|with location,
|and suggests
|redoing the
|f3 action

<h5977>
f3cardidisstx  1200
               activepause
               kb

lk             ki
10             sh
24
bx             cleansetopbox.

<h5978>
spaceaddtexts= |at pos, txt2
|In:pos,txt1,  |is added
|txt2          |{can be &&}
|Txt1 receives |after spaces
|& MUST be
|roomy, it's   sx
|extended w/   tx
```

```
|spaces then,  s3

<h5979>
jx             ex
lk
s4             ix
               lk
i3             m3
n?             ad
               jx
se             setlenandnil

<h5980>
|From:         |Qty:
ix             ix
up             lk
               |Handles also
|To:           |pos < len
jx             |of txt1
i3
ad             fw

<h5981>
i4             i3
i3             f4
ge             su

               jx
se             i4
               ad
ex             ns.

<h5982>
isfieldfresh=  warp4x135
|In: x, y      10
|Gives: flag   wk
|Action: Finds |typenum is
|out whether   |conclusive
|the 4x135     |enough
|field's fresh
|{ie, empty}   n?.

<h5983>
finddoclen=    135
|Gives: num
|Action: finds |Note, an as
|the highest   |yet unfilled
|numbered row  |spreadsheet
|{max 135}     |gives basis
|with content
|in it         s4

<h5984>
```

```
ll:2000000000  i4
ll:4           ex

i2
i4
isfieldfresh

d2             lo

<h5985>
q4             twobillion
               s1


i4             lo
ye

d2             i4.

<h5986>
buildrowstr=   t5
|In:txt,pos,
|x,y,fnwarp to t4
|fcminterface  |txt must have
|Builds txt w/ |room
|this 4x135    t3
|field tabulat t2
|to given pos  t1

<h5987>
j2             j3
               j4
               j5
               stdquotepos



j1             spaceaddtexts.

<h5988>
putlnnum=      makenumber
|In:number
|Action: via
|putlntxt
|stores the
|number on a
|line;assumes
|initialized   putlntxt.

<h5989>
getlinenum=    getlinetext
|Gives:num,    sx
|flag
```

```
|Action:       intonum
|Exactly as    |Assumes initd
|getlinetext   |getlinetext
|but converts
|quote=>num    ix.

<h5990>
maxlen=        f
|In:txt,maxlen lk
|Gives:txt     ix
|Crops when    le
|txtlen>maxlen
sx             se
tx
jx             ex

<h5991>
ix


jx



setlenandnil.

<h5992>
putpairaya=    |inclusive,1st
|In: fn#       |of the pair,
|Action: uses  |ie, num1,
|putlnnum to   |is nil {eg,
|put num1<cr>  |like "type"}
|num2<cr>      |Assumes initd
|from fnaya    |putlntxt
|array up to,& sx

<h5993>
ll:2000000000  i5
|<cr>, ie,     putlnnum
|lineshift,    |include also
|after each    |completenil
i1             f1
ix             ix
fnaya          fnaya
s5             putlnnum

<h5994>
i5             h1
n?

se
```

```
ex             lo.

<h5995>
getpairaya=    |inclusive,1st
|In: max,fn#   |of the pair,
|Action: uses  |ie, n1,is nil
|getlinetext   |--or until
|to get n1<cr> |arraylen=max
|n2<cr> pairs  |is reached
|into fnaya    sx
|array up to,& s9

<h5996>
ll:2000000000  |include nil
getlinenum     h1
sh             getlinenum
s5             |ok here
i5             sh
i1             i1
ix             ix
fnyay          fnyay

<h5997>
f1             se
i9
ge             ex

i5
n?

or             lo.

<h5998>
texttoyay=     sx
|In: text,fn#  s9
|Action: uses  i9
|fnyay to put  lk
|text incl     s5
|initial len &
|nilchar into
|fnaya array

<h5999>
ll:2000000000  m1
               i5
m1             gt
i9
ay             se
i1
ix             ex
fnyay          lo.
```

```
<h6000>
putfielddata=  s8
|In:x y        s9
|Action: uses
|putlntxt,
|assumed to be i9
|initiated, to i8
|store all of  warp4x135
|a ss field    tx

<h6001>
& &            i8
s3             makenumber
i9             s3
96             i3
ad             1
f3             insertachar
kl             i3
i3             putlntxt

<h6002>
|Type#         10
|1=num         jx
|2=text        wk
|3=formula     f
|decisive      s5
|for storage
|{only 3 has
|formulacomp}  putlnnum

<h6003>
|Recent value, 12
|relevant for  jx
|two of the    wk
|types but
|easy to
|include
|anyhow
|yes          putlnnum

<h6004>
37             i5
jx             1
wk             eq

|Qty           se
|decimals

putlnnum       ex

<h6005>
i5             jx
2              link2tr1warp
```

```
eq              putlntxt
n?              |Within a fn:
                |Fieldtexts
                |type2 maxlen
                |14
d4              ex

<h6006>
|Type 3         sx
|formula,       ix
|first text     quotefnaya
|form from      |maxlen ca 70
|array l#2:     |so one quote
51              |easily holds
jx              |it
wk              putlntxt

<h6007>
|Then the       sx
|pairs of       ix
|{type,val}
|until the
|nilcomplete
52
jx
wk              putpairaya.

<h6008>
savethismany=  t5
|In:qtyrows,   s9
|disk#,card#,  s8
|fnwrpinterfac sx
|Gives:qtycard |Will also
|Saves row#1   |accept save
|to qtyrows in |of fresh
|qtycards      |spreadsheet

<h6009>
i8              |groupsignuse?
i9              16
initputlntxt    j5
|The 'code-     wk
|word':
^SpreadSheet

putlntxt        putlnnum

<h6010>
                |groupsign is
                |comma?
                numgroupsign
                44
                eq
```

```
               putlnnum

<h6011>
ll:2000000000  okputlntxt
ix             theputlncrd
               lk
i1             i9
               su
ge

d6             ex

<h6012>
ll:4           i2
               i1
               putfielddata
i2
i1
isfieldfresh
               lo
d3             lo.

<h6013>
cleansedoc=    ll:135
|In:fnwarp to  ll:4
|fcminterface  |resets <f8>;
|Spreadsheet   |but marker xy
|RESET FOR ALL |left intact
|FIELDS, for   i2
|ctrt&refresh  i1
tx             databas4x135

<h6014>
i2             0
i1             12
warp4x135      ix
sx             kw
0
10
ix
kw

<h6015>
               0
               37
               ix
               kw


               lo
               lo
```

```
<h6016>
decimalsign    |Markerupdate
44             |will normally
eq             |be called
               |with 1 1 as
se             |the newx,newy
               |as its input
               |after any
togglenumsign  |cleanse

<h6017>
basis          jx
16             l1maintodance
jx
kw
dance
31
jx
kw             sayctrtactive.

<h6018>
getfielddata=  warp4x135
|In:x, y
|Gives:more?
|Action:Uses
|getlinetext;
|more?=can be
|more to come;
|to x y field  sx

<h6019>
getlinenum     getlinenum
s3             s3
s5             12
               ix
i5
10
ix
kw             kw

<h6020>
getlinenum     gt
s3             an
37
ix             d2
kw
i3
i5             i3
1              ex

<h6021>
getlinetext    51
```

```
s3             ix
               wk

74
maxlen
|For both text
|and formula   texttoyay

<h6022>
i3             gt
               an

               d2


i5             i3
2              ex

<h6023>
|Then the      74
|pairs of      j5
|{type,val}    getpairaya
|until nil     |Can
52             |Always be
ix             |More
wk             dance.
t5

<h6024>
loadsaveddoc=  t5
|In:disk#,     s9
|card#, fnwarp s8
|fcminterface  basis
|Gives:qtycard
|Loads a saved
|spreadsheet   j5
|document      cleansedoc

<h6025>
i8             n?
i9             ix
linetextstart  n?
               or
getlinetext
sx             se
^SpreadSheet   |At stk:basis
te             ex

<h6026>
getlinenum     se
sx
16
j5
```

```
kw

ix
n?             ex

<h6027>
getlinenum     ix
sx             n?
n?
               se
se
               ex

togglenumsign  sh

<h6028>
ll:2000000000  i5
getlinetext    n?
sx             or
parseposname   ix
s5             n?
s4             or
i4
n?             t2

<h6029>
j2             i4
               i5
               getfielddata




d4             sx

<h6030>
j2             thelinecrd
n?             lk
ix             u2
an             i9
               su

               ex
d6             lo.

<h6031>
xportthismany= t5
|In:qtyrows,   s9
|disk#,card#,  s8
|fnwrpinterfac sx
|Gives:qtycard |Will also
|All of row#1  |accept xport
|to qtyrows in |of fresh
```

```
|qtycards=>b9  |spreadsheet

<h6032>
f3quote1       0
lk             jx
tx             setlenandnil
i8
i9             |Pr column:
initputlntxt   14
               t1
ll:2000000000  ll:4

<h6033>
i2
i1

isfieldfresh



dh

<h6034>
jx             i2
tn             i1
m2             tn
j1             tn
mm             tn
up             j5
tn             tn
tn             buildrowstr

<h6035>
lo             okputlntxt
jx             theputlncrd
putlntxt       lk
i1             i9
ix             su
lt             ex

d6             lo.

<h6036>
f3savetx=      haps i1} ===>
^.
               *txtcomplete
longtxt*       cliptrail
OVERWRITES!!!
Please type ri
ght location t f3savetx
o save to {per kl

<h6037>
```

```
f3save=        tx
|In:x,y,fnwarp
|fcminterface
|Saves entire  s8
|spreadsheet
typedinput
lk
sx             s9

<h6038>
               f3savetx



               lk
               10
               24
               bx

<h6039>
905            ci
23
8              s6
ix             s5
l9
ix
cliptrail
clipleading    cleansetopbox

<h6040>
i5             tellf3cidiss
ye
i6             i9
ye             i8
an             jx
               markerupdate

d6             ex

<h6041>
finddoclen     j5
i5
i6
jx

savethismany

t5             tellf3success

<h6042>
1              4
allrownums     1
               jx
               markerupdate
dance          1
37             1
jx             jx
kw             markerupdate.

<h6043>
f3loadtx=      ADSHEET IN RAM
^.             ===>
               *txtcomplete
longtxt*       cliptrail
Place of saved
 spreadsheet,
eg i1? NOTE:TH f3loadtx
IS CLEARS SPRE kl

<h6044>
f3load=        tx
|In:x,y,fnwarp
|fcminterface
|Loads saved   s8
|spreadsheet
typedinput
lk
sx             s9

<h6045>
               f3loadtx



               lk
               10
               24
               bx

<h6046>
905            ci
23
8              s6
ix             s5
l9
ix
cliptrail
clipleading    cleansetopbox

<h6047>
i5             tellf3cidiss
ye
i6             i9
ye             i8
an             jx
               markerupdate
```

```
d6             ex

<h6048>
               j5
i5
i6
jx

loadsaveddoc

t5             tellf3success

<h6049>
1              4
allrownums     1
               jx
               markerupdate
dance          1
37             1
jx             jx
kw             markerupdate.

<h6050>
f3exporttx=     eg c9000 ===>
^.
               *txtcomplete
longtxt*       cliptrail
Export {1-way}
 to B9edit of
sums etc, pls  f3exporttx
type location, kl

<h6051>
f3export=      tx
|In:x,y,fnwarp
|fcminterface
|Exports to    s8
|b9edit format
typedinput
lk
sx             s9

<h6052>
               f3exporttx



               lk
               10
               24
               bx
```

```
<h6053>
905            ci
23
8              s6
ix             s5
l9
ix
cliptrail
clipleading    cleansetopbox

<h6054>
i5             tellf3cidiss
ye
i6             i9
ye             i8
an             jx
               markerupdate

d6             ex

<h6055>
finddoclen     j5
i5
i6
jx

xportthismany

t5             tellf3success

<h6056>
1              4
allrownums     1
               jx
               markerupdate
dance          1
37             1
jx             jx
kw             markerupdate.

<h6057>
keyf3menutx1=  EXPORT result
^.             to B9EDIT
               *txtcomplete
longtxt*       cliptrail
MENU:<1>SAVE {
backup often}
<2> LOAD {as s keyf3menutx1
aved by 1} <3> kl

<h6058>
keyf3=         tx
|in:x,y,fnwarp
```

```
|to interface
|Subroutine of s8
|fcminterface,
|processes
|press of
|<f3> key      s9

<h6059>
cleansetopbox  keyf3menutx1
               lk



               10
               24
               bx

<h6060>
ki
48
su
sx



cleansetopbox

<h6061>
ix             i9
1              i8
eq             jx
n?             f3save


               jx
d6             rewriteallpos

<h6062>
ix             i9
2              i8
eq             jx
n?             f3load


               jx
d6             rewriteallpos

<h6063>
ix             i9
3              i8
eq             jx
n?             f3export
```

```
               jx
d6             rewriteallpos

<h6064>
10             |The reason we
jx             |fetch the
wk             |marker x y
12             |anew is that
jx             |load always
wk             |likes to
jx             |reaffirm the
markerupdate.  |new start! :)

<h6065>
setdnumissutx= he formula. Cf
^.             r manual. :)
               *txtcomplete
longtxt*       cliptrail
Please: do era
se a formula,
then use this, setdnumissutx
 then retype t kl

<h6066>
setdnumissue=  setdnumissutx
|Action: tells
|to pls change
|only non-
|formula field lk
               10
               24
               bx

<h6067>
1200
activepause
kb
ki
sh


cleansetopbox.

<h6068>
setdnumthistx= W etc} ==>
^.
               *txtcomplete
longtxt*       cliptrail
Pls type qty o
f decimals {cf
r docs for num setdnumthistx
ber sizes, DIV kl
```

```
<h6069>
setdnumthis=   xyto4x135
|In:x,y,fnwarp warp4x135
|fcminterface  tx
|Sets pos4x135
|qty decimals  |What type?
typedinput     10
lk             jx
sx             wk

<h6070>
|Type3=formula setdnumthistx
3
eq
n?
d2             lk
               10
setdnumissue   24
ex             bx

<h6071>
955            intonum
23             cleansetopbox
1
ix             0
l9
ix             8
cliptrail      makefit
clipleading    s5

<h6072>
37             scaleint
jx             12
wk             jx
i5             kw
               i5
12             37
jx             jx
wk             kw.

<h6073>
setdnumalltx1= LL OVER? Y/N <
^.             ENTER>=Y
               *txtcomplete
longtxt*       cliptrail
Only affects f
ormulae made A
FTER this. Cha setdnumalltx1
nge decimals A kl

<h6074>
setdnumalltx2= r sizes, DIVW
```

```
^.             etc} ==>
               *txtcomplete
longtxt*       cliptrail
Global setting
 of qty of dec
imals {cfr doc setdnumalltx2
s as for numbe kl

<h6075>
maybednumcha=  s3
|in:qty,x,y    s2
|action:       s1
|subroutine in
|setdnumall,
|changes
|qty decimals
|for nonformul

<h6076>
               10
               jx
               wk
i2
i3
warp4x135
               3
tx             eq

<h6077>
se             37
               jx
               wk
ex



               i1

<h6078>
12             i1
jx             37
wk             jx
scaleint
12
jx

kw             kw.

<h6079>
setdnumall=    setdnumalltx1
|action:asks;
|then sets qty
|decimals all
```

```
|nonformfields lk
typedinput     10
lk             24
sx             bx

<h6080>
955            parseyenter
23             n?
3
ix
kb             se
l9
cleansetopbox
ix             ex

<h6081>
cleansetopbox


               setdnumalltx2
               lk
               10
               24
               bx

<h6082>
955            intonum
23             cleansetopbox
1
ix             0
l9
ix             8
cliptrail      makefit
clipleading    s5

<h6083>
ll:135         lo
ll:4


i5
i2
i1
maybednumcha   lo.

<h6084>
toggroupntx1=   any key,eg <S
^.             PACE>, thanks
               *txtcomplete
longtxt*       cliptrail
Toggling group
ing of digits
{eg 1,234,567} toggroupntx1
```

```
 on/off: press kl

<h6085>
toggroupnums=  tx
|In: fnwarp
|fcminterface  toggroupntx1
|Action:       lk
|toggles flag
|#3 connected  10
|to use of     24
|groupsigns    bx

<h6086>
kb             16
               jx
               wk
ki
sh             n?
               16
               jx
cleansetopbox  kw.

<h6087>
togcommadtx1=  "," use. Pls p
^.             ress <space>
               *txtcomplete
longtxt*       cliptrail
When you use n
umbers like 99
,999.00, this  togcommadtx1
switches "." & kl

<h6088>
togcommadot=   tx
|In: fnwarp
|fcminterface  togcommadtx1
|Action:       lk
|toggles
|global comma  10
|and dot       24
|settings      bx

<h6089>
kb


ki
sh


cleansetopbox  togglenumsign.

<h6090>
```

```
keyf8menutx1=  1,234 <4>switc
^.             h comma & dot
               *txtcomplete
longtxt*       cliptrail
MENU:<1>decima
ls this field
<2>all over <3 keyf8menutx1
>group digits  kl

<h6091>
keyf8=         tx
|in:x,y,fnwarp
|to interface
|Subroutine of s8
|fcminterface,
|processes
|press of
|<f8> key      s9

<h6092>
cleansetopbox  keyf8menutx1
               lk



               10
               24
               bx

<h6093>
ki
48
su
sx



cleansetopbox

<h6094>
ix             i9
1              i8
eq             jx
n?             setdnumthis


               jx
d4             rewriteallpos

<h6095>
ix             setdnumall
2
eq
```

```
n?


               jx
d3             rewriteallpos

<h6096>
ix             jx
3              toggroupnums
eq
n?


               jx
d4             rewriteallpos

<h6097>
ix             jx
4              togcommadot
eq
n?


               jx
d4             rewriteallpos

<h6098>
i9
i8




jx
markerupdate.

<h6099>
keyctrt=       tx
|in:fnwarp
|to interface
|Subroutine of
|fcminterface,
|processes
|press of
|ctr-t

<h6100>
31             sayctrtactive
jx             jx
flipgram       l1maintodance
               ex
               sayctrtslash
               ex
               tn
d4             tn.

<h6101>
fcminterface=  |#10:row# top
|in:tr#,fnwarp |#1:scrx;scry
|Action: show  |#5:flag cls?
|& edit whole  |#8:flag ctrt:
|spreadsheet   | #5 assumes 1
tx             |#9:compvars
sh             |#3:groupsign?
|triplet-uses: |l#1:refreshfn

<h6102>
jx             jx
fetchl1main    |the #5 flag
n?             |called 'cls'
|The #9 tripl  |is typically
|for comppos & |only used in
|qtydecimal by |the startup
|'compileform'
d2             refreshallpos

<h6103>
31             jx
jx             |The tr#3 flag
wk             |says: should
|ctr-t flag    |numgroupsign
s9             |be employed,
i9             |eg 1,234,567,
               |all fields
d2             l1maintobasis

<h6104>
               10
               jx
               wk
               s3
               12
               jx
               wk
               s4

<h6105>
22             spreadscreen
jx
solvegram

n?
               i3
               i4
d4             posclsblack
```

```
<h6106>
advancedck     i9
n?             an

|Up,beneath,   se
|right,left--
|273, 274,
|275, 276;
|keycodes      ex

<h6107>
ki             q4
sx
273
ix             i3
eq             i4
n?             jx
               markerupdate
d6             ex

<h6108>
274            h4
ix
eq
n?             i3
               i4
               jx
               markerupdate
d6             ex

<h6109>
275            h3
ix
eq
n?             i3
               i4
               jx
               markerupdate
d6             ex

<h6110>
276            q3
ix
eq
n?             i3
               i4
               jx
               markerupdate
d6             ex

<h6111>
ix             i3
278            i4
```

```
|Ascii:<home>
eq
n?             jx

               keyhome
d5             ex

<h6112>
ix             i3
13             i4
|Ascii:<enter>
eq
n?             jx

               keyenter
d5             ex

<h6113>
ix             jx
280
|Ascii:<pgup>  keypgup
eq
n?


d3             ex

<h6114>
ix             jx
281
|Ascii:<pgdn>  keypgdn
eq
n?


d3             ex

<h6115>
ix             i3
277            i4
|Ascii:insert
eq
n?             jx

               keyinsert
d5             ex

<h6116>
ix             i3
284            i4
|Ascii:<f3>
eq
n?             jx
```

```
               keyf3
d5             ex

<h6117>
ix             i3
289            i4
|Ascii:<f8>
eq
n?             jx

               keyf8
d5             ex

<h6118>
ix             jx
20
|Ascii:<ctr-t> keyctrt
eq
n?


d3             ex

<h6119>
ix             elegantleave
27
|Ascii:<esc>
eq
n?


d2             ex.

<h6120>
twobillion
setfundlevel

               basis
               3999
&fcminterface& basis
3999           &spreadsheet&
fnactcherish   fneasyact

<h6121>
dance          dance
5              10
tripletpos     adjustfund
adjustfund
dance
8              dance
tripletpos     12
adjustfund     adjustfund
```

```
<h6122>
dance
10
tripletpos
adjustfund
|Triplet#10
|tells which
|rownumber
|is at y=1

<h6123>
|The refreshfn |qty links:
|is checked    1
|by all the    47
|spreadsheet   adjustfund
|positions,    &refreshfn&
|when flag 1,  fnam
|full recalc   50
|takes place   adjustfund

<h6124>
now=           450
               250
               af
^spreadsheet   ad
pp             activepause
^starting! *** fcm
pp             |Optionally:
               qu.

<h6125>
               &now&






               zz





*****
FINIS
```