ＪＡＰＬＡ蓼科夏の合宿資料　2006/8/5

Ｊ－ＯＯＰ　Ｇｒｉｄのプログラミング　その６
Ｊ－ＧｒｉｄＳｈｅｅｔ
Ｅｘｃｅｌライクだが、Ｅｘｃｅｌより強力！

西川　利男

Ｊ－Ｇｒｉｄプログラミングによる数独パズルをこれまで何回か紹介してきた。しかし、Ｊ－Ｇｒｉｄの本来の目的はもちろんＥｘｃｅｌと同様の表計算にある。　Ｊ－ＧｒｉｄＳｈｅｅｔ（Ｊによる表計算システム）はＥｘｃｅｌよりもずっと融通性があり、かつ強力であることをお目にかけたい。

１．Ｊ－ＯＯＰとＪ－ＧｒｉｄＳｈｅｅｔ

Ｊ－Ｇｒｉｄプログラミングは ＪのＯＯＰ（オブジェクト指向プログラミング）の手法に従って、プログラミングされる。最近のプログラミング言語はＯＯＰレベルのものが多いが、これが中々分かりにくい。それどころか、何ゆえに「オブジェクト指向」なるパラダイムが現在もてはやされるのか、単なる流行か、それなりに価値を持つものか？……あまり良く知られていない感がする。筆者はＪのＧｒｉｄの適用というＯＯＰの実践を通してその有り難味を体得できた。さらに「数独パズル」という格好のテーマがあったことも良かった。今回、あらためてＪ－ＯＯＰ　Ｇｒｉｄを見直してみたいと思う。

２．Ｊ－ＧｒｉｄＳｈｅｅｔとＥｘｃｅｌ

一般に表計算ソフト（Spread Sheet）では、マウスなどで複数のセルを選択し、それらのデータをまとめて演算できることが最大の特徴である。

その際、２つの場合がある。

(1) 合計、平均などの計算

(2) セル要素のそれぞれの値に対して平方根、対数などの計算

Ｅｘｃｅｌでは(1)の場合，結果は自動的につぎのセルに入れられる。(2)の場合は先に結果を入れるセルを指定しなければならないなど面倒である。Ｅｘｃｅｌでは一つのセルに値と計算式とが同居しているため何かと問題が多い。

われわれのＪ－ＧｒｉｄＳｈｅｅｔでは、このようなことのないよう、計算の結果を確認し、その格納のセルをマウスで指定するようにした。また、計算式についてもＪの命令そのものが使用できるようにした。その結果、例えば「素数を取り出す」といった計算もいと易く行える。これらにより、Ｅｘｃｅｌなどよりはるかに融通性のある処理システムとなっている。

３．Ｊ－ＧｒｉｄＳｈｅｅｔの実際

run '' として起動すると次のようにセルで区切られたシート画面が現れる。

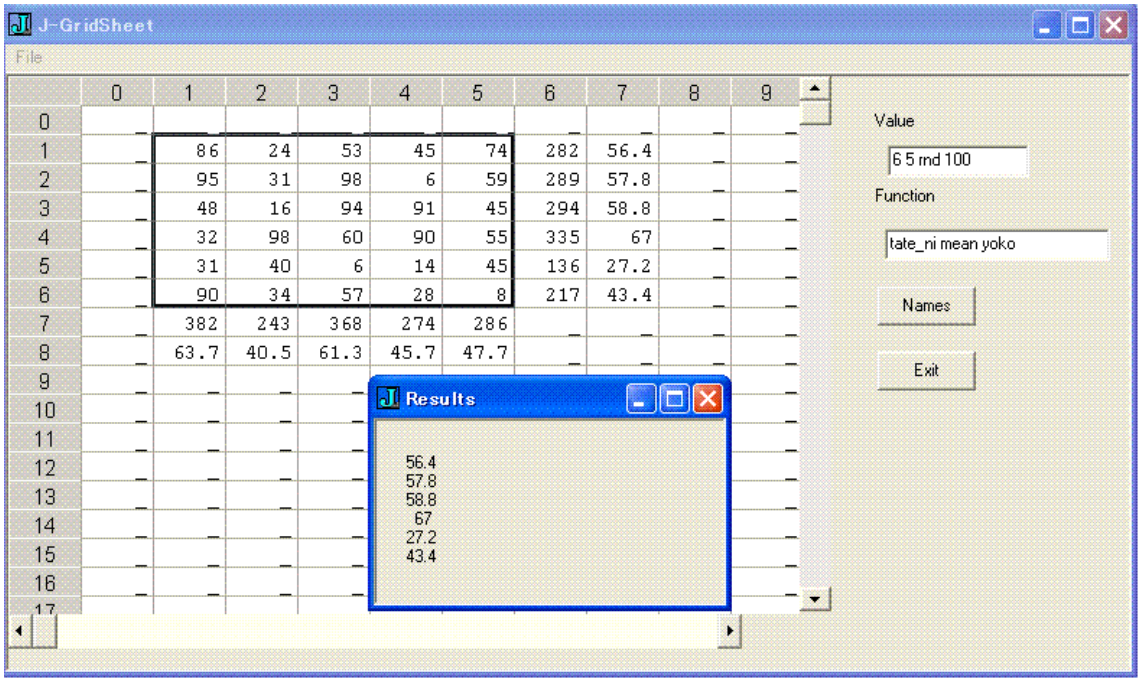
３．１　タテ・ヨコの集計表

乱数のデータを入れて、いろいろな計算をやってみる。まず **Value** 入力窓にたとえば

6 5 rnd 100

と打ち込むと、乱数データが作られる。ここで格納したいセルの位置を左上として、マウスの右ボタンを押すと（または左ボタンのダブルクリックにより）、値が貼り付けられる。

次にマウスの左ボタンのドラッグで範囲を選択し、**Function** 入力窓に sum と入れる。（もちろん +/ と入れることもできる。）合計の結果が表示されるので、これをデータのセルの下に右ボタンにより貼り付ける。続いて同様によりドラッグにより選択し、**Function** 入力窓には mean と入れれば平均が得られる。これはさらに下に貼り付ける。



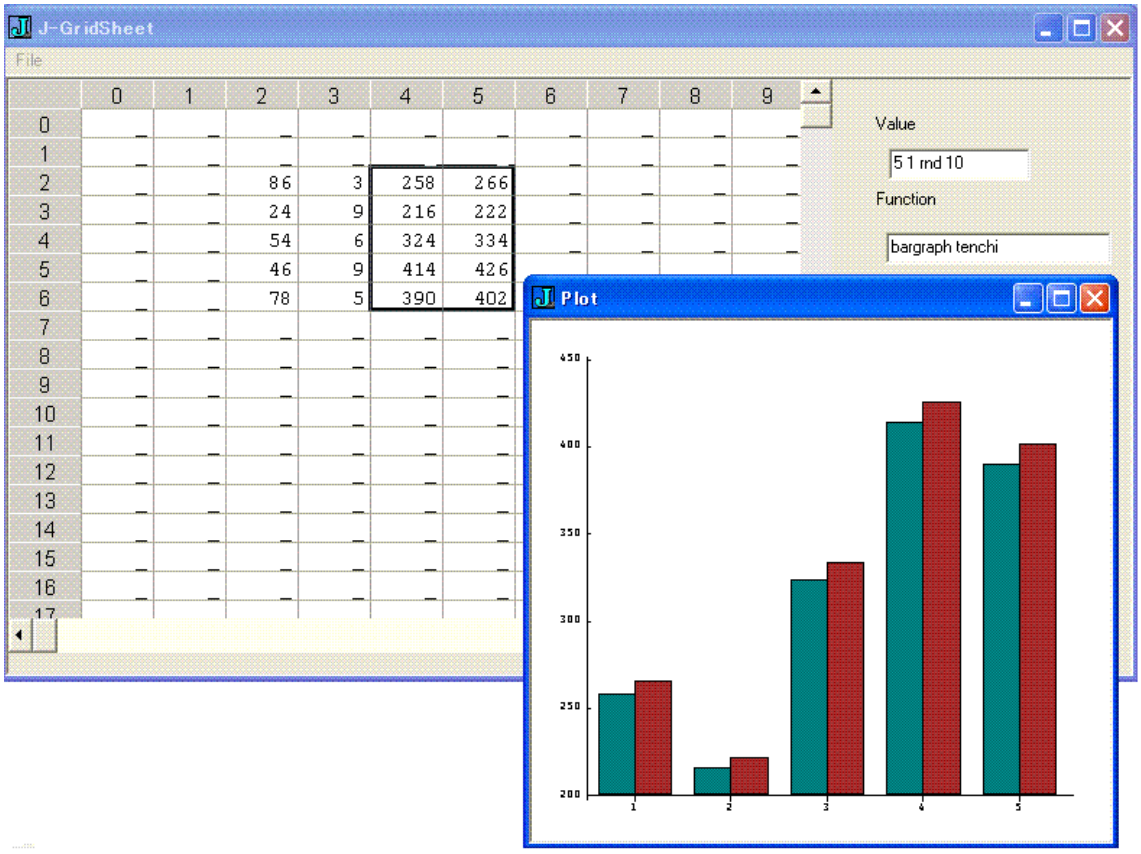
	0	1	2	3	4	5	6	7	8	9
0										
1		86	24	53	45	74	282	56.4		
2		95	31	98	6	59	289	57.8		
3		48	16	94	91	45	294	58.8		
4		32	98	60	90	55	335	67		
5		31	40	6	14	45	136	27.2		
6		90	34	57	28	8	217	43.4		
7		382	243	368	274	286				
8		63.7	40.5	61.3	45.7	47.7				
9										
10										
11										
12										
13										
14										
15										
16										

今度はマウス・ドラッグ選択の後、**Function** 入力窓に sum yoko と入力すればヨコ方向の合計が得られるがこのままでは具合が悪い。**Function** 入力窓には tate_ni としてから、データの右側に貼り付ける。これを一度で行うには、tate_ni sum yoko とすればよい。ヨコ方向の平均も求めて隣に貼り付ける。ここまでの画面のようすは以上のようになる。

３．２　買い物の計算とグラフ表示

まず、品物の単価と個数を入れる。この２列のセルを選択してから、**Function** 入力窓には tate_ni prod yoko と入れると、それぞれの金額が得られるので、ヨコに貼り付ける。次にこのセルを選択して、**Function** 入力窓に 1.03 * と入力すると、税金を含めた金額が計算されるので、これも隣に貼り付ける。今度はこれらの結果をグラフ表示してみよう。

それにはこの２列のセルを選択してから、**Function** 入力窓には bargraph tenchi と入れれば、ただちにグラフが描かれる。なお、グラフ化には配列の転置が必要である。ここまでのようすは以下のようになる。



３．３　素数の計算

Value 入力窓には >: i. 100 のように入力すれば、１から１００までの整数が作られる。これを選択して、**Function** 入力窓には prime と打ち込むだけで、素数が取り出される。

いうまでもなく、J-GridSheet の中では素数を求める関数 prime は J のプリミティブ q: を使って次のように定義されている。

```
prime =: ((1: = #@q:) #])@,
```

もちろん、つぎのような直接の J プログラムを使うことも可能である。[*)]

```
primes =: 3 : '(2 = +/ 0 = n |/ n) # n =. , y.'
```

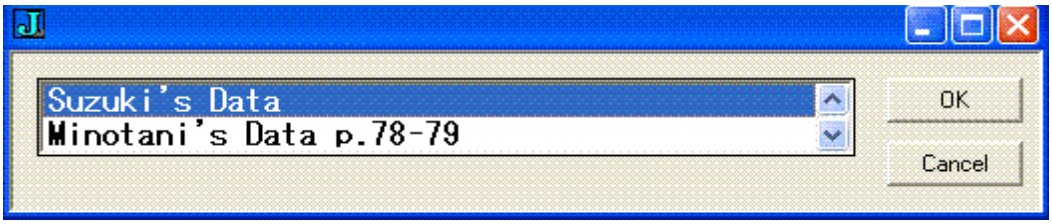
[*)] 西川利男「基礎からのＡＰＬ」サイエンスハウス、p.104 （APL から J に書き直した）

３．４　外部ファイルからの取り込みと統計計算表の作成

先月のＪの研究会で鈴木義一郎氏から次の論文発表があった。

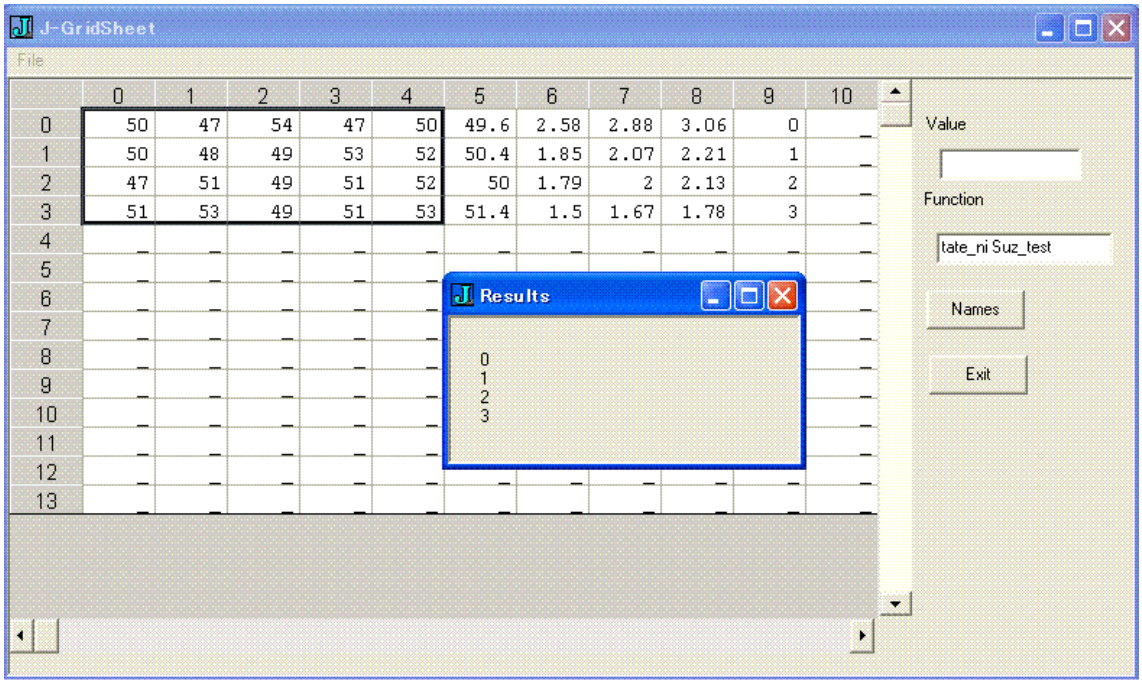
鈴木義一郎「標本分散と不偏分散」

この値を外部ファイルデータとして読み込み、同じ計算を J-GridSheet でもやってみた。ファイルデータで起動するには、open '' と打ち込む。すると下のような選択ボックスが開くので、望むデータをマウスで選び色を替え、ＯＫボタンを押す。



ここでは、Suzuki's Data を選んで、起動すると、セルには標本データが入れられたシート画面が現れる。

そこで、鈴木論文と同じような統計表を作ってみよう。まず、セル全体をマウスドラッグで選択し、前と同様に **Function** 入力窓には tate_ni mean yoko と入れ、右ヨコに貼り付ける。



ここで、それぞれの統計関数はつぎのような名前で定義してある。

標本分散と標準偏差　(S^2, S)　　s_var　s_dev

不偏分散とその平方根　(s^2, s)　　u_var　u_dev

分散と標準偏差の不偏推定(s_0^2, s_0)　k_var　k_dev

鈴木の判定テスト　　Suz_test

したがって、再びセル全体を選択し、**Function** 入力窓には tate_ni s_dev yoko と入れ、そのヨコに貼り付ける。同様にして次々に、u_dev, k_dev の計算を行って、最後には **Function** 入力窓には　tate_ni Suz_test として貼り付ける。

このようにして統計表が上のように出来上がった。一度に計算した結果を示すのではなく、計算しながら順次表示して行くという過程はこのようなスプレッドシート形式によってのみ可能であり、その教育的な効果が大きいと思われる。

なお、いままでにも出てきたが、J のいろいろな命令、関数などはわかり易くするため、名前による定義を行っている。そしてこれらの一覧は、[Names] ボタンを押して見ることが出来る。

４．Ｊ－Ｇｒｉｄ Ｓｈｅｅｔのプログラム

Ｊによるプログラムは、以下の２つのプログラム

J_GridSheet.ijs

pgridtest.ijs

から成り立ち、その詳細は最後にあげてあるが、かなり大きい。

夏季合宿では、ＪのＯＯＰ、Ｇｒｉｄの基本から、プログラムのコーディングに即して丁寧に説明したいと考えている。

５．参考資料

筆者のこれまでの発表は以下のようになっている。

その１　ＪのＯＯＰとは－簡単な例でやってみる　　JAPLA シンポジウム 2005/12/10

その２　Ｊのスプレッドシートと数独パズルへの適用 JAPLA シンポジウム 2005/12/10

その３　Ｊの Grid でカレンダーを作って遊ぶ　　Ｊ研究会資料　2006/2/25

その４　Ｊ - Grid による数独パズルをもっと使いやすく　Ｊ研究会資料　2006/3/25

その５　Ｊによる数独パズル－棋譜の自動記録と棋譜データによる実行　Ｊ研究会資料 2006/6/24

また、Ｊの [Studio] － [Labs] にある以下の詳細なチュートリアルも参照されたい。

- Grid Basic Examples
- Grid Control
- Grid Low Level Programming
- Object Oriented Programming
- Locales

付録　Ｊ－ＯＯＰ　Ｇｒｉｄに関連した操作命令

・セル内に文字列を書き込む

glgridrchw i, j, m, n　　セル位置 i, j から範囲 m, n を指定する

glgridtext DATA　　m × n コの文字列を書き込む

・ウィンドウ画面を更新表示する

glpaintx ' '

・セル内の属性（色、ワク、文字フォント）を変える

glgridatt , atts ' '　　属性テーブルを読み込む

glgridrchw i, j, m, n　　セル位置 i, j から範囲 m, n を指定する

glgridtype , m, n $ type_N　属性（type 0, type 1, type 2）を設定

・マウス左ボタンのクリックでセル位置を得る

mouse_jzgrid_ sysdata

'i j' =: mark_jzgrid_　　セル位置 i, j を得る

・マウス左ボタンのドラッグでセル範囲を得る

'ix, jx' =: extent_jzgrid_　セル範囲 ix, jx を得る

・セル内の文字列を得る

DA =: glgridgettext i,j

・デバッグのためにクラス内の文字列を直接書き出す

smoutput DA

・対話ボックスに文字列を書き出す

wdstatus DA

・対話ボックスからリスト選択を行う

wdselect DA

メインプログラム

```
NB. J-GridSheet = Excel-like Spreadsheet on J
NB.   by T. Nishikawa, 2006/6/26
NB. using class ptestgrid.ijs

corequire 'user\classes\ptestgrid'
run =: 3 : 0
d1 =: 20 20$_  NB. for blank cells
NB. d1 =: (10#8) ? 100 NB. Random values
NB. Pascal Triangle Numbers
NB. pas =: 0&, + ,&0
NB. d1 =: pas^:(i.20) 1
w1 =: 'd1' conew 'ptestgrid'
)

NB. open file from stat01.txt,  2006/7/27
require 'files'
open =: 3 : 0
DAT =: 'm' fread 'stat01.txt'
id =: ((65&<: *. 122&>:)  a. i. {."(1) DAT) # i.#DAT
id =: id, #DAT
DAT =: DAT, ' '
ix =: > {: wdselect <"(1) id{ DAT
sta =: >: ix{id
end =: <: (>:ix){id
X =: (sta + i.>:end-sta) { DAT
X =: ". X
'R C' =: $X
d1 =: ((R+10), (C+10)) $_
d1 =: X (<(i.R);(i.C))} d1
w1 =: 'd1' conew 'ptestgrid'
)
```

クラスプログラム

```
NB. class ptestgrid
NB. using jwatch, jwgrid, jzgrid
NB. applied to J-GridSheet system

coclass 'ptestgrid'
corequire 'jwatch'
corequire 'jwgrid'
corequire 'jzgrid'

PTESTGRID=: 0 : 0
pc ptestgrid;pn "J-Grid";
menupop "File";
menu new "&New" "" "" "";
menu open "&Open" "" "" "";
menusep ;
menu exit "&Exit" "" "" "";
menupopz;
xywh 317 65 34 12;cc go button;cn "Names";
xywh 318 85 34 12;cc cancel button;cn "Exit";
xywh 0 0 300 166;cc grid isigraph;
xywh 0 166 299 11;cc sb scrollbar;
xywh 301 0 11 165;cc sbv scrollbarv;
xywh 320 47 62 11;cc func edit ws_border es_autohscroll;
xywh 316 34 50 10;cc Function static;
xywh 317 11 50 10;cc label static;cn "Value";
xywh 321 21 50 11;cc value edit ws_border es_autohscroll;
pas 6 6;pcenter;
rem form end;
)

create=: 3 : 0
9!:11 (3)  NB. display 2 decimal point
wd PTESTGRID
formhwnd=: wd'qhwndp'
NB. initialize form here
```

```
grid =: '' conew 'jwgrid'
sizeenable__grid =: 1
init__grid 'd1_base_';'grid';'sb';'sbv'
DA =: d1_base_ NB. same as dataname__grid
wd 'pn *J-GridSheet'
Res =: ''
wd 'pshow;'
)

destroy=: 3 : 0
wd'pclose'
codestroy''
wdstatus ''
''
)

ptestgrid_cancel=:ptestgrid_cancel_button=:ptestgrid_close=:destroy

formselect=: 3 : 'wd''psel '',formhwnd'

ptestgrid_sb_button=: 3 : 0
scrollbar__grid sb
wd 'setfocus grid'
)

ptestgrid_sbv_button=: 3 : 0
scrollbarv__grid sbv
wd 'setfocus grid'
)

ptestgrid_grid_size=:        3 : 'size__grid 0'

NB. =========================================================
NB. Advanced Excel_type Calculator by T.N. 2006/3/6 - 7/25
NB. (1) Select cells by Mouse Dragging, or "Value" Edit_Box
NB. (2) Enter J Function in "Function" Edit_Box
```

```
NB.  eg. +/ <- sum, (+/ % #) <- mean
NB.      +: <- double, -: <- half
NB.      *: <- square, %: <- square root
NB. (3) Position by Double Click, Calculated results diplayed
NB. Calculated results are written into Clipboard
NB. Then, if editenable OK by CTRL-e
NB. Results can be pasted into cells by CTRL-v
NB. Named Functions
double   =: +:
half     =: -:
square   =: *:
sqrt     =: %:
rnd        =: 4 : '(({.x.) # ({: x.)) ? y.'
sum        =: +/
prod   =: */
mean     =: +/ % #
sqd    =: sum@(*:@[ -"(1 0) mean))
s_var  =: sqd % #
s_dev  =: sqrt @ s_var
u_var  =: sqd % <:@#
u_dev  =: sqrt @ u_var
kn     =: 3 : '2 * square (! <: (# y.) % 2) % (! <: ((#y.) - 1) % 2)'
k_var  =: sqd % kn
k_dev  =: sqrt @ k_var
Suz_test =: 3 : '+/"(_1) 4.02 > (s_var, u_var, k_var)"1 y.'

clear     =: (%&0`(_"_))@.(0&=)"(0)
yoko   =: "(1)
tate_ni =: ,.
tenchi  =: |:
prime    =: ((1: = #@q:) #])@,

graph  =: 3 : 0
   require 'plot'
   plot y.
)
```

```
bargraph =: 3 : 0
   require 'plot'
   'bar' plot y.
)



NB. array in string convert 2006/7/17
NB.    ar_str 'i. 2 3' => (2,3)$0 1 2 3 4 5
NB.    ar_str 'i. 3'   => 0 1 2
NB.    ar_str '123'    => 123
ar_str =: 3 : 0
V =. ". y.
select. #$V
      case. 0 do. 'R C' =. 1, 1
      case. 1 do. 'R C' =. 1, $V
      case. 2 do. 'R C' =. $V
end.
S0 =: (": V),"(1)' '
if. 1 < , R
   do.
      S =. '(',(":,R),',',(":,C),')$', (,S0)
   else.
      S =. ,S0
   end.
}: S
)

NB. Input Values on Mouse_Drag
ptestgrid_grid_mbldown=: 3 : 0
   mbldown__grid sysdata
NB.  smoutput mouse__grid sysdata
   'R C' =: mark__grid
NB.  smoutput <: R, C
)
ptestgrid_grid_mmove=:   3 : 'mmove__grid sysdata'
```

```
ptestgrid_grid_mblup=:   3 : 0
   mblup__grid sysdata
   'RX CX' =: extent__grid
   NB. SDATA are Values in String Reprsentation
   SDATA =: '': (<(<:R+i.RX);(<:C+i.CX)){DA
   smoutput #SDMN =. $'': SDATA
   if. 1 = #SDMN do. SDATA =: (SDMN, 1)$SDATA end.
   smoutput 'Data:'
   smoutput '': SDATA
   'Result' wdstatus ''                    NB. revised at long last
   'Data' wdstatus '': SDATA
)

NB. Input Values on Value_Edit Button
ptestgrid_value_button=: 3 : 0
NB. value == entered by Value_Edit
SDATA =: ar_str value
Res =: '': SDATA
smoutput '': SDATA
'Result' wdstatus ''
'Data' wdstatus '': SDATA
)

NB. Enter Function on Function_Edit Button
ptestgrid_func_button=: 3 : 0
F =. func
smoutput 'Values:'
SDD =. ar_str SDATA
smoutput '': SDD
smoutput 'Function: ', F
NB. Res is Values in Number
Res =: '': F ,' ', SDD
if. 'clear' -: F
  do. (R, C) paste Res
        return.
  end.
```

```
smoutput 'Results:'
smoutput Res
'Data' wdstatus "                              NB. revised at long last
NB. 'Results' wdstatus DA_RC$' '  NB. revised at long last
'Results' wdstatus Res
SDATA =: '': Res
wd 'clipcopyx *', clipfmt__grid ('': Res)   NB. writes in clipboard
)



NB. Calculated Values into Cell by Double Click
celldata=: 3 :'';(':each y.),each 0{a.'
ptestgrid_grid_mbldbl=:  3 : 0
   mbldbl__grid sysdata
   'R C' =: mark__grid
NB.  paste Res on cells by 'Double_Click'
   (R, C) paste Res
)
NB. Calculated Values into Cell by Mouse Right Button Down
NB.   2006/7/26
ptestgrid_grid_mbrdown=: 3 : 0
   mbldbl__grid sysdata
   'R C' =: mark__grid
NB.  paste Res on cells by 'Right_Down'
   (R, C) paste Res
)

NB. Paste on cells (R, C) paste Res
paste =: 3 : 0
:
'R C' =. x.
Res =. y.
   if. 0=#Res do. return. end.
   select. #$Res
      case. 0 do. 'M N' =: 1, 1
      case. 1 do. 'M N' =: 1, $Res
```

```
      case. 2 do. 'M N' =: $Res
   end.
   glgridrchw R, C, M, N
   glgridtext celldata Res
   glpaintx ''
   DA =: Res (<(<:R+i.M);(<:C+i.N)) } DA
)



ptestgrid_go_button=: 3 : 0
   Messages =: 'sum, prod, mean, ',LF, 'yoko, tate_ni, tenchi', LF, 'clear'
   Messages =: Messages,  LF, 'double, half, square, sqrt, rnd, prime'
   Messages =: Messages,  LF, 's_var, s_dev, u_var, u_dev, k_var, k_dev, Suz_test'
   Messages =: Messages,  LF, 'graph, bargraph'
   'Defined Names:' wdstatus Messages
)

NB. =========================================================

ptestgrid_grid_char=:    3 : 'char__grid sysdata'
ptestgrid_grid_copy=:    3 : 'copy__grid 0'                NB. CTRL-c
ptestgrid_grid_paste=:   3 : 'paste__grid 0'   NB. CTRL-v



ptestgrid_ectrl_fkey=: 3 : 0
editenable__grid=: -.editenable__grid
)
```