

```

to in x w y h a
  (: x . : w . : y . : h .
    (x <
      (mouse 8)
      <
      (x + w))
    ?
    (y <
      (mouse 9)
      <
      (y + h)))
to butlon
  (!
    (4 = mouse 4))
to but2on
  (!
    (2 = mouse 2))
to but3on
  (!
    (1 = mouse 1))
to frredit frm
  (: frm . repeat
    (butlon ?
      (
        (in frm x 8 frm y 8)
        ?
        (frm fclear . wait(butloff)
          . wait(butlon)
          . white . frm frame . black . frm param
            (" winx _ " frm x _
              (mouse 8)
              | 16 . " winy _ " frm y _
                (mouse 9))
          . frm fclear . frm frame . frm show .)
          (in frm x +
            (frm wd - 8)
            8 frm y +
            (frm ht - 8)
            8)
          ?
          (frm fclear . wait(butloff)
            . wait(butlon)
            . white . frm frame . black . frm param
              (" winwd _ " frm wd _
                (
                  (
                    (mouse 8)
                    - frm x)
                  | 16)
                + 16 . " winht _ " frm ht _
                  (mouse 9)
                  - frm y)
            . frm fclear . frm frame . frm show .)
          (in frm x 8 frm y +
            (frm ht - 8)

```

```

      8)
      ?(window)
      (in frmx + frmwd - 8 8 frmy 8)
      ?
      (white . frm frame . black . frm clear . sched remove frm))
      kbdchk ?
      (fev frm)
      done))
to showedit func
  (: # func . newdisp . edit func .)
to butloff
  (
    (4 = mouse 4)
    ?
    (! false)
    ! 1)
to wait x
  (: " x . repeat
    (
      (x eval)
      ?(done)))
to fev disp
  (: disp . cr . read eval print)
to init
  (addto dispframe "
    (% mouse ?
      (
        (in frmx frmwd frmy frmht)
        ?
        (SELF fclear . SELF show . SELF frame . SELF _ 13 . SELF _ 32 . repe
at
      (
        (in frmx frmwd frmy frmht)
        ?
        (blink SELF . frmedit SELF)
        done))))
  addto number "
    (% | ?
      (!
        (SELF / : x)
        * x))
    . " sched _ schedule 40 . . disp fclear . disp param
    (" winht _ " frmht _ 14 . " winy _ " frmy _ 656)
    . disp frame . disp _ "Scheduler Running!" sched _ disp . window . sched run
  .)
to so sz
  (: sz . do 4
    (go sz right 90))
to schedule : ints i j k m
  (isnew ?
    (" ints _ vector : . " i _ " j _ 0)
    % _ ?
    (for j to ints length by 2
      (null ints [ j ] ?
        (ints [ j ] _ : . ints [ " j _ j + 1 ] _ " mouse . done)

```

```

    0 = ints [ j ] ?
    (ints [ j ] _ : . ints [ " j _ j + 1 ] _ " mouse . done)))
% run ?
    (repeat
      (ints eval))
% print ?
    ('schedule' print . " i _ 0 . for k to ints length by 2
      (null ints [ k ] ?(again)
        " i _ i + 1)
      . sp . i print . sp . " items print)
% remove ?
    (: m . for i to j by 2
      (eq ints [ i ] m ?
        (ints [ i ] _ 0 . ints [ i + 1 ] _ 0 .)))
% look ?
    (ints print))
to nkbd
  (1 CODE 20)
to newdisp disp
  (" disp _ dispframe 16 496 1 400 string 2000 . show func)
to keys : buf t
  (isnew ?
    (" buf _ 0 - 1)
  % active ?
    (
      (" buf _ 1 CODE 20)
      < 0 ?
        (! false)
        ! 1)
    buf < 0 ?
      (! DRIBBLE next _ translation [ TTY ])
    " t _ buf . " buf _ 0 - 1 . ! DRIBBLE next _ translation [ t ])
to window p
  (" p _ dispframe 192 192 200 256 string 1000 . p frame . p _ 'This is a windo
w - more later' . sched _ p)
to blink d
  (: d . d _ 20 . d _ 21 . do 100()
  . . d _ 8 . d _ 8 .)
to kbdchk
  (1 CODE 20)
to dvect vtr i j k
  (: vtr . " j _ 1 . repeat
    (
      (" k _ sscan vtr [ j to vtr length ] find first " ?)
      > 0 ?
        (for i = j to k + 1(vptit)
          . cr . " j _ k + 2)
        for i = j to vtr length(vptit)
          done))
to vptit
  (vtr [ i ] is vector ?
    (" $ print)
    vtr [ i ] print)

```