```
* scribe.s
* 2001-06-16
* Daniel Lidström
* danli97@ite.mh.se
* www.ite.mh.se/~danli97/
::
 RECLAIMDISP                            clear display
 TURNMENUOFF                            turn off menu
CODE
* status flags to signal increase or decrease
sINCX     EQU       0
sINCY     EQU       1
* start position
XSTART    EQU       65
YSTART    EQU       32
          GOSBVL    =SAVPTR             save pointers
          INTOFF                        turn off maskable interrupts
          GOSBVL    =D0->Row1           A[A] & D0 -> start of screen
          R0=A                          save in R0[A]
          C=0       A
          LC(2)     XSTART              x
          B=C       A                   B[A] = x coordinate
          LC(2)     YSTART              y
          D=C       A                   D[A] = y coordinate
          ST=1      sINCX               set movement to up-right
          ST=0      sINCY
loop      C=D       A                   calculate 34*y
          C=C+C     A
          A=C       A
          CSL       A
          A=A+C     A                   A[A] = 34*y
          C=B       A                   calculate no. of horizontal nibbles
          CSRB.F    A
          CSRB.F    A                   C[A] = hor. nibbles
          C=C+A     A
          A=R0                          get screen address
          C=C+A     A
          D0=C                          D0 -> correct nibble in screen
          C=B       A                   compute pixel mask
          A=C       A
          LC(1)     3
          A=A&C     P
          C=0       A
          C=C+1     A                   assume pixel mask is 1
          A=A-1     P                   done?
          GOC       +   >-----+         yes, write pixel
          C=C+C     P         |         no, shift left one
          A=A-1     P         |         done?
          GOC       +   >-----+         yes, write pixel
          C=C+C     P         |         no, shift left one
          A=A-1     P         |         done?
          GOC       +   >-----+         yes, write pixel
          C=C+C     P         |         no, shift left one
+         A=DAT0    P   <-----+         read current nibble
          A=A!C     P                   OR in new pixel
          DAT0=A    P                   write new nibble to screen
          B=B+1     A                   assume x is increasing
          ?ST=1     sINCX               assumption correct?
          GOYES     +     >---+         yes
          B=B-1     A         |         no, decrease two
          B=B-1     A         |         to compensate +1
          GONC      ++    >---|-+       no carry means x>=0
          B=0       A         | |       x is -1
          B=B+1     A         | |       set x to 1
          ST=1      sINCX     | |       signal x increasing now
          GOTO      ++    >---|-+       test y
+         LC(2)     130   <---+ |       check max x
          ?C>B      A           |       x below max?
          GOYES     ++    >-----+       yes, test y
          ST=0      sINCX       |       signal x decreasing now
++        D=D+1     A     <-----+       comments here are the same as for x
          ?ST=1     sINCY
          GOYES     +
          D=D-1     A
          D=D-1     A
          GONC      ++
          D=0       A
          D=D+1     A
          ST=1      sINCY
          GOTO      ++
+         LC(2)     63
          ?C>D      A
          GOYES     ++
          ST=0      sINCY
++        LCHEX     1FF                 load all keys
          OUT=C
          GOSBVL    =CINRTN             read keys
          ?C#0      B                   got any keys?
          GOYES     exit                yes, exit
          GOTO      loop                no, continue loop
exit      INTON                         turn on maskable interrupts
          GOVLNG    =GETPTRLOOP         restore rpl pointers, exit
ENDCODE
 FLUSH                                  flush keyboard
;
```