```
********* T E C O   C O M M A N D S *********
DEFALUT VALUE OF NUMERIC ARGUMENTS IN [ ]:
  c1=Ø, c2=Z, 1=1(except in S, :S, R, :R), c=Ø, n=1,
  r=1(except in < ... >), w=1

        ERs   External Read
      1 EWs   External Write
[c1,c2] EWs
      1 EOs   External Overwrite
[c1,c2] EOs
    [c] J     Jump to the position c
    [n] C     advance n Characters
    [1] L     advance 1 Lines
    [1] :L    1L-C
    [1] T     Type out
  c1,c2 T
[11[,12]] V   View: equivalent to '1-11T :T/$$/ 12T'
              default: 11=1, 12=11
    [n] D     Delete n characters
    [1] K     Kill
  c1,c2 K
    [1] :K    Kill one less(1>Ø) or one more(1<=Ø) newline than 1K does.
        Is    Insert string
  [r,]e I     Insert chr(e) r times
[[1,]n] Ss    Search n times within the range 1 (if specified)
[[1,]n] :Ss   Search and return signal: success...-1; fail...Ø
[[1,]n] Rs1s2  Replace
[[1,]n] :Rs1s2 Replace and return signal
    [1] Xq    eXtract string from the buffer
  c1,c2 Xq
        Gq    Get string from Q-register q
        :Iqs  Insert s into Q-register q
        :Ts   Type s on the terminal
  [r,]e :T    Type chr(e) r times
    [x] Uq    Update: x=9999999 unless specified
    x,y Uq    do yUq and return x
    [r] < ... >  execute ... r times: r=9999999 unless specified
      f ;     exit from the loop when f>=Ø
        =     type out a newline
      x =     type out x followed by a space
    x,w =     write(x:w)
        ET    External cTime
        EQ    External Quit
[x[,y]] Mqs1s2...ss  Macro call
        :Xq   get the next string argument and store it into q
        [q    push
        ]q    pop
      x "C ... :' .... '  if chr(x) is alphanumeric then ... else .... fi
                          (else part(:' ....) is optional.)
  x[,y] "E ... :' .... '  if x=y then ... else .... fi
                          (y=Ø unless specified.)
                          ("N --> x<>y;  "G --> x>y;  "L --> x<y;
                           ":G --> x>=y; ":L --> x<=y)
        Os    gO to s
        !label!  label or comment
     c' A     return character code: not Ascii but EBCDIK
        #     read decimal number to the right of .
  x[,w] #     insert decimal representation
        :Wq   Wait for a line from the terminal
[x[,y]] W     Wipe out its arguments
              (\, $, and newline have the same effect as W.)
        ?     trace on
        ??    trace off
        @     errorset mode on
        @@    errorset mode off

***** NUMERIC ARGUMENTS *****
c1,c2  absolute range
1      (1) top of 1'th line from the current position
           e.g. 1=-1...previous line, 1=Ø...current line, 1=1...next line
       (2) range between the current position and the location
           specified by 1(1)
c      absolute location
c'     relative location
n      (direction) * (iteration factor)
         direction = 1 ...forward, direction = -1 ...backward
r      iteration factor (r>=Ø)
w      width of the field
e      EBCDIK code (Ø <= e <= 255)
f      exit condition (exit when f>=Ø)
x,y    no special meaning
```