```
IF
 X?0 or X?Y or Flag(n) = True (provide flag number on next instruction)
   do something
ELSE                                  (optional)
   do something else
ENDIF

note: no special keycode for 'flag(n)=true', just use KC_O to KC_7

SF(n): To set flag to true, with 'n' provided as 'next instruction'
n between 0 and 7 in a 8 bit flag register
CF(n): To set flag to false, with 'n' provided as 'next instruction'

CASE
 var to test (0 means X, 1 means Y, ...)
WHEN
 value 1 (integers only) then optional KCP_NUMSEP keycode (#) (only required if next stmt starts with a digit)
   do something                              automatically convert speace or CR to 'NumSep' when keyed from PC
WHEN                   1+ 'when' stmts       present as Carriage Return (CR) on Calculator LCD
 value 2                                     optionally can show # as well on computer
   do something else
OTHER                         other' is optional and must be at end
   do another thing
ENDCAS

FOR
 CNT_INIT i (counter identifier)
 init count (16 bit signed integer)
COUNT?
 n (signed integer) then optional # (NumSep)
   Do stuff
NEXT
 increment / decrement to count (signed integer) then optional #

CND_INIT i (optional)
WHILE
 X?0 or X?Y or Flag(n) = True
   do stuff
   CNT_INC i n # (optional, to increment Cnt(i) by n)
WEND

DO
   do stuff
LOOPWH
 X?0 or X?Y or Flag(n) = True

EXIT instruction: force loops to return
CYCLE instruction: to force loops to loop

CNT_REF by itself, will push CNT(i) to X on the stack

@@ single line comment
{- multiple line comment -}
~~A to mark beginning of program A
```