```
/*
 *  Find the matching token, or matching bracket.
 *
 *  Based on a KEDIT macro by Michael Chase.
 *
 *  With []{}()<>, works like CMATCH.
 *
 *  Have the cursor on the first character when matching a
 *  multi-character token:
 *
 *        do    end      #if   #endif     if  fi    etc.
 *        ^     ^        ^     ^          ^   ^
 *
 *  Assign to a key; the key currently used for CMATCH is a good choice.
 *
 *  The token pairs are made available by executing the LMATCHES macro.
 *
 *
 */

if incommand() then exit
if eof() | tof() then exit

/* select search tokens */
'extract /column/line/field/'
match = translate(substr(field.1, field.3))
do pair = 1
   'editv getf matches.'pair
   first = matches.pair
   pair  = pair + 1
   'editv getf matches.'pair
   second = matches.pair
   if first = "" | second = "" then do
      /* Check for single character tokens */
      if pos(field.2, "[]{}()<>") \= 0 then do
         /* regular match characters */
         'cmatch'
         exit rc
         end
      nop
      exit 2
      end

   if first = substr(match, 1, length(first)) then do
      match   = first
      matcher = second
      way     = "+"
      leave
      end
   else if second = substr(match, 1, length(second)) then do
      match   = second
      matcher = first
      way     = "-"
      leave
      end
   end

'extract /thighlight/'
'set thighlight off'

'extract /field/'
```

```
lenf  = length(field.1)
lenr  = length(matcher)
head  = " "d2c(9)
tail  = head";.,"
level = 0
r     = 0
col   = column.1()
do forever
   if way = "-" then col = col - 1
   else col = col + 1
   do forever
      if col < 1 | col > lenf then do
         'nomsg locate 'way'\'matcher'\ | 'way'\'match'\'
         if rc \= 0 then do
            r = rc
            ':'line.1
            'cl:'column.1
            exit r
            end
         'extract /field/'
         lenf = length(field.1)
         if way = "-" then col = lenf
         else col = 0
         end

      call get_cols way col matcher
      col1 = ca
      cole = cb
      if col1 \= 0 then do
         if pos(substr(field.1, cole, 1), tail) = 0 then col1 = 0
         else
         if col1 > 1 then
            if pos(substr(field.1, col1 - 1, 1), head) = 0 then col1 = 0
         end

      call get_cols way col match
      col2 = ca
      cole = cb
      if col2 \= 0 then do
         if pos(substr(field.1, cole, 1), tail) = 0 then col2 = 0
         else
         if col2 > 1 then
            if pos(substr(field.1, col2 - 1, 1), head) = 0 then col2 = 0
         end

      if col1 = 0 then col = col2
      else if col2 = 0 then col = col1
      else if way = "-" then col = max(col1, col2)
      else col = min(col1, col2)
      if col > 0 then leave
      end
   'cl:'col
   if translate(substr(field.1, col, lenr)) = matcher then level = level - 1
   else level = level + 1
   if level < 0 then leave
   end

'cl:'col
'set thighlight 'thighlight.1
exit
```

```
get_cols: procedure expose ca cb
/* ca: first column of token,   */
/*      or 0 if token not found */

/* cb: last column of token + 1 */
/*      junk if token not found */
parse arg w c m

if w = '+' then
   ca = pos(m, translate(curline.3()), c+1)
else
   ca = lastpos(m, translate(curline.3()), c)

   cb = ca + length(m)

return
```