NAME

HEBWID − Approximate the typeset width in points of the Hebrew string described by CHASH.

SYNOPSIS

W=HEBWID(CHASH,VHASH,NC,PT)

W	is the REAL*4 width estimated
CHASH(NC)	is the INTEGER*1 vector of letter hash codes
VHASH(NC)	is the INTEGER*1 vector of vowel or trope hash codes
NC	is the INTEGER*1 number of letters specified by CHASH
PT	is the REAL*4 Redis point size that will be used

WARNING

The number NC is, perhaps surprisingly, a 1-byte integer rather than a fullword.

DESCRIPTION

This routine contains a table giving the width of 50 adjacent characters of each Hebrew "consonant" defined in HB2HSH, when it is typeset at 17-point size. Using each element of CHASH as an index into this table, the routine looks up the widths x 50 of the characters specified and accumulates their sum.

If the VHASH code corresponding to a character is for the \hebspc{}{} command, then XYLOAD is used to find how many 15ths of a \hem it specifies and that fraction of 50\hem is added to the sum.

Finally, the sum is divided by 50 and scaled by the actual point size for return.

SEE ALSO

STRWID, which approximates the printed width of an English string.

DIAGNOSTICS

If the value of a hash code is out of the range [1,40] and thus does not correspond to a Hebrew character identified in HB2HSH, then the value returned for W is the negative of the index in CHASH where the offending letter appears.

LINKAGE

```
gfortran source.f −L${HOME}/lib −lmisc
```

AUTHOR

Michael Kupferschmid

EXAMPLE

```
      CHARACTER*39 HSTR/'\hebrew{lamed}\qamats{coph}\hebrew{yod}'/
      INTEGER*1 NC,CHASH(3),VHASH(3)
      INTEGER*4 RC
      CALL HB2HSH(39,HSTR,3, NC,CHASH,VHASH,RC,K)
      S=HEBWID(CHASH,VHASH,NC,17.0)
      PRINT *,S
      STOP
      END
```

This example produced the following output:

```
unix[1] a.out
   24.959999
unix[2]
```

Printing the string with a rule of this width beneath it shows that the approximation is accurate.