NAME

STRWID − Approximate the typeset width in points of the string in STRING.

SYNOPSIS

W=STRWID(STRING,LS,PT)

W	is the REAL*4 width estimated
STRING(LS)	is the CHARACTER*1 string whose typeset width is to be estimated
LS	is the INTEGER*4 index of the last character in STRING
PT	is the REAL*4 Computer Modern point size that will be used

DESCRIPTION

This routine contains tables of the lower-case letters, upper-case letters, decimal digits, and printable other characters. Corresponding to each is a table of the width in points of a string consisting of 50 of each character when it is printed at 12-point size. The widths of these 50-character strings were determined experimentally using the ${HOME}/bin/widths.tex LaTeX document. For each character in STRING the routine finds the index of the character in one of the tables and adds the corresponding width to an integer total. This sum of the character widths is then divided by 50 and scaled by the point size for return. The point sizes PT corresponding to the different standard LaTeX font sizes are shown in the table below. These were determined experimentally using the /home/mike/man/test/strwidtest.f program. For other font sizes see the table in /home/mike/bin/fontsizes. Using the nominal point sizes of 6 for \tiny, 8 for \scriptsize, 10 for \footnotesize, 11 for \small, and 14 for \large yields lengths that are significantly wrong.

\tiny	7.5
\scriptsize	8.65
\footnotesize	10.15
\small	11.2
\normalsize	12.0
\large	14.4

SEE ALSO

HEBWID, which approximates the printed width of a Hebrew string.

DIAGNOSTICS

If an input character is not found in the tables, the value returned for W is the negative of the index in STRING where the offending character appears.

BUGS

The widths of some non-alphabetic symbols are estimated because they are difficult to measure. Not all non-alphabetic symbols are included.

LINKAGE

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

AUTHOR

Michael Kupferschmid

EXAMPLE

```
      S=STRWID('Hello There testing',19,12.0)
      PRINT *,S
      STOP
      END
```

This example produced the following output:

```
unix[1] a.out
   99.160004
unix[2]
```

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