

NAME

LISTRN – Tokenize the Hebrew transliterations in a line of text.

SYNOPSIS

CALL LISTRN(LINE,L,MXTL, TLC,NTL)

LINE(*)	is the input character string
L	is the INTEGER*4 length of LINE
MXTL	is the INTEGER*4 second dimension of TLC
TLC(18,MXTL)	is the CHARACTER*1 list of transliterations returned as columns
NTL	is the INTEGER*4 number of transliterations found, or -1 if an error occurred

DESCRIPTION

This routine searches LINE from right to left for a substring that looks like <transliteration>. When it finds one it stores the transliteration as a column in TLC. It continues searching and adding substrings to TLC until all of LINE has been examined. Because the line is parsed from right to left, the beginning of a transliteration is marked by > and its end by <.

DIAGNOSTICS

If the routine returns with NTL=-1, the cause of the error will be identified by one of the following messages printed on standard-error.

> encountered while in transliteration
too many transliterations on this line
< encountered outside transliteration
incomplete transliteration

LINKAGE

gfortran source.f -L\${HOME}/lib -lmisc

AUTHOR

Michael Kupferschmid

EXAMPLE

```
CHARACTER*17 LINE/'<abc> stuff <def>'/
CHARACTER*1 TLC(18,100)
CALL LISTRN(LINE,17,100, TLC,NTL)
WRITE(6,901) NTL
901 FORMAT('found these',I3,' transliterations')
DO 3 I=1,NTL
    L=LENGTH(TLC(1,I),18)
    WRITE(6,902) (TLC(K,I),K=1,L)
902    FORMAT(18A1)
3 CONTINUE
STOP
END
```

This example produced the following output:

```
unix[1] a.out
found these  2 transliterations
def
abc
unix[2]
```