NAME

STRINS − Insert one character string into another.

SYNOPSIS

CALL STRINS(STRING,LS,INSERT,LI,LOC,JOB, RESULT,LR)

STRING(LS)	is the CHARACTER*1 template
LS	is the INTEGER*4 dimensioned length of STRING
INSERT(LI)	is the CHARACTER*1 string to insert in STRING
LI	is the INTEGER*4 index of the last nonblank in INSERT
LOC	is the INTEGER*4 index in STRING at which to start inserting
JOB	is an INTEGER*4 flag telling how to space the result; see table below
RESULT(LS)	is the CHARACTER*1 result string, the same length as STRING
LR	is the INTEGER*4 index of the last nonblank in RESULT

JOB	spacing
0	leave spacing unchanged
1	leave one blank at the end of the inserted INSERT
2	leave no blanks at the end of the inserted INSERT

WARNING

The return parameter LR must not be given as a literal on input. The dimensioned size of RESULT is LS. The same variable name must not be passed for LI and LR. The routine assigns LR=LS before LI is used, so passing the same actual parameter for LI and LR aliases LI to LS. A run-time error of exceeding the dimensioned size of INSERT results from making this mistake.

DESCRIPTION

If LS ≤ 0 or LOC ≤ 0 or LOC > LS, the routine returns immediately. Otherwise, it copies STRING into RESULT and inserts as much of INSERT as will fit between LOC and the end of RESULT. Then it edits the spacing according to the value of JOB, as described in the table above. Finally it finds LR, the index of the last nonblank in RESULT. If the template does not need to be preserved, STRING and RESULT can be the same variable; if they are not the same variable, then STRING is left unchanged. However, as noted above LS and LR MUST NOT BE THE SAME VARIABLE. Absent parameter aliasing, LS, INSERT, LI, LOC and JOB are never changed.

SEE ALSO

INTINS, which inserts an integer into a character string

LINKAGE

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

AUTHOR

Michael Kupferschmid

EXAMPLE

```
      CHARACTER*1 MSG(23),X/'x'/
      CALL STRINS('The filename is "     "',23,X,1,18,0,MSG,LM)
      CALL MESAGE(MSG,LM)
      CALL STRINS('The filename is "     "',23,X,1,18,1,MSG,LM)
      CALL MESAGE(MSG,LM)
      CALL STRINS('The filename is "     "',23,X,1,18,2,MSG,LM)
      CALL MESAGE(MSG,LM)
      STOP
      END
```

This example produced the following output:

```
unix[1] a.out
The filename is "x    "
The filename is "x "
The filename is "x"
unix[2]
```