RelString Compare two Pascal-style strings for sort order

#include <OSUtils.h> **Operating System Utilities**

short	**RelString(***strA*, *strB*, *caseSens*, *diacSens*); [128K ROMs]	
Str255	*strA* ;	Pascal-style strings to compare
Str255	*strB* ;	
Boolean	*caseSens* ;	should upper/lowercase count?
Boolean	*diacSense* ;	should diacritical marks count?
	returns	-1: *strA* **<** *strB*; 0: equal, 1: *strA* **>***strB*

RelString compares two Pascal-style length-prefixed strings (optionally ignoring case and/or diacritical marks), and returns an indication of which comes first in the ASCII collating sequence.

strA and . . .
strB are addresses of Pascal-style length-prefixed strings.

caseSens specifies whether or not the comparison should be case-sensitive. It must be one of:

FALSE ignore character case when comparing ('A' == 'a')
TRUE character case is significant ('A' != 'a')

diacSens specifies whether or not the comparison should be sensitive to diacritical marks. It must be one of:

FALSE ignore diacritical marks when comparing ('å' == 'a')
TRUE diacritical marks are significant ('å' != 'a')

Returns: a signed integer; it indicates the relative collating value of the strings, considering the case- and diacritical-sensitivity. Using the same values as with the familiar strcmp() library function, it is one of:

-1 *strA* is less than *strB*
0 *strA* is equal to *strB*
+1 *strA* is greater than *strB*

Notes: Since **RelString** compares Pascal-style strings directly, it is handier than converting to C-style strings and using strcmp.

If *caseSens* =FALSE, then both strings are treated as if they had been upshifted with **UprString** (though the original contents are not modified).

For 64K ROMs, the **EqualString** function can be used to test if two strings are the same. The **IUEqualString** and **IUCompString** functions take into consideration special spelling conventions used in foreign languages.