SAY --Subprogram

Format
CALL SAY(word-string[,direct-string][,...])

Cross Reference
SPGET

Description
The SAY subprogram enables you to instruct the computer to produce speech.

Word-string is a string-expression whose value is any of the words or phrases in the computer's resident vocabulary. If you use a string constant, you must enclose it in quotation marks. Alphabetic characters must be upper-case.

The computer substitutes "UHOH" for a word-string not in the vocabulary.

A speech phrase (more than one word) must be enclosed in pound signs(#). A speech phrase must be predefined; that is it must be resident in the computer's vocabulary.

A compound is a new word formed by combining two words already in the vocabulary. For example, SOME+THING produces "something" and THERE+FOUR produces "therefore". A compound must not be enclosed in pound signs.

See Appendix H for a list of the computer's resident vocabulary .

Direct-string is a string-expression whose value is the computer's internal representation of a word or phrase. You can use or modify a direct-string returned by the SPGET subprogram.

See Appendix I for information on adding suffixes to direct-strings. You can specify multiple word-strings and direct-strings by alternating them. To specify two consecutive word-strings or direct-strings, enter an extra comma as a separator between them.

Examples
100 CALL SAY("HELLO, HOW ARE YOU")
Causes the computer to say "Hello, how are you".

CALL SAY(A$,,B$)
Causes the computer to say the words indicated by A$ and b$, which must have been returned by SPGET.

The following program illustrates a use of CALL SAY with a word-string and three direct-strings.

100 CALL SPGET("HOW",X$)
110 CALL SPGET("ARE",Y$)
120 CALL SPGET("YOU",Z$)
130 CALL SAY("HELLO",X$,,Y$,,Z$)