POKEV --Subprogram--Poke to VDP RAM
POKEV

Format
CALL POKEV(address,byte-list[,"",address,byte-list[,...]])

Cross Reference
LOAD,PEEK,PEEKV,VALHEX

Description
The POKEV subprogram enables you to assign values directly to specified VDP memory addresses.

You can use the LOAD subprogram to assign values to CPU.

The address is a numeric-expression whose value specifies the first VDP(Video Display Processor) memory address where data is to be poked. If the byte-list specifies more than one byte of data, the bytes are assigned to sequential memory addresses starting with the address you specify.

The address must have a value from 0 to 16383 inclusive.

If you know the hexadecimal value of the address (>0000->3FFF), you can use the valhex function to convert it to a decimal numeric-expression.

If necessary, the address is rounded to the nearest integer.

The byte-list consists of one or more bytes of data, separated by commas, that are to be poked into VDP memory starting with the specified address.

Each byte in the byte-list must be a numeric-expression with a value from 0 to 32767. If the value of a byte is greater than 255, it is repeatedly reduced by 256 until it is less than 256. If necessary, a byte is rounded to the nearest integer.

You can specify multiple addresses and byte-lists by entering a null string(two adjacent quotation marks) as a separator between a byte-list and the next address.

If you call the POKEV subprogram with invalid parameters the computer may function erratically. If this occurs, turn off the computer, wait several seconds, then turn the computer back on.

Examples
100 CALL POKEV(3333,233)
Pokes the value 233 into location 3333.

100 CALL POKEV(13784,273)
Pokes the value 17 (273 reduced bye 256 once) into location 13784.

100 CALL POKEV(7343,246,"",VALHEX("2E4F"),433)
Pokes the value 246 into location 7343, and uses VALHEX to ascertain the decimal value equivalent of the hexadecimal number 2E4F (11855). The value 177(433 reduced by 256 once) is then poked into this location.