

Pascal always passes strings by address, since strings' lengths may vary.

Pointers:

Address pointers are unsigned integers that occupy 1 word of storage. Their format is identical to that of integers, except that their values may range from 0 to 65535. The value of a pointer, in this implementation of Pascal, is the memory address of the object being described.

Example: The address of AN0 (one of the annunciator ports) is hex C058 (49240 decimal). This address is stored as:

```

MSB 1  1  0  0  0  0  0  0  0  1  0  1  1  0  0  0 LSB
    <----->  <----->  <----->  <----->
          C          0          5          8

```

Pointers, like integers, may be passed by value or by reference (as a Var parameter).

Long Integers:

Long integers are a special type of variable, first defined at UCSD as part of their extensions to the Pascal language. They are primarily used to handle calculations involving numbers which (a) cannot be represented accurately in floating point (real) format, and (b) are too large to store in integer format.

Long integers are stored in BCD (binary coded decimal)--one digit per nybble. One entire word is reserved for the sign of the long integer, and the variable must end on a word boundary. Four digits can be contained in one word, so the smallest definable long integer takes up two words of memory. The numbers are padded with leading zeroes when necessary to fill up the last word. The sign is 0 if positive and 255 if negative. (One byte is used for the sign.)

To illustrate the structure of long integers, let's take a specific example: the long integer -123456 takes 3 words: one for the sign, and two for the digits (since they are stored in multiples of 4). This long integer is stored in the following format:

```

<----- each digit is one nybble ----->
MSB 6  5  4  3  2  1  0  0  0  0  F  F LSB
    <-- word -->    <-- word -->    < sign word >

```

A long integer should always be passed by address, since its length depends on its definition.

Apple Tech Notes

Tech Info Library Article Number:688