Referencing an Array

To reference a specific element of an array, you must use subscripts. Subscripts are numeric-expressions enclosed in parentheses immediately following the reference to the array-name. An array must include one subscript for each dimension in the array. If necessary, the value of a subscript is rounded to the nearest integer.

Reserving Space for Arrays

When you use DIM as a program statement, the computer reserves space for arrays when enter the RUN instruction, before your program is actually run. If the computer cannot reserve space for an array with the dimensions you specify, the message Memory Full in line-number is displayed, and the command does note execute.

When you use DIM as a command, if the computer cannot reserve space for an array with the dimensions you specify, the message Memory Full is displayed and the command does not execute.

Until you place values in an array, each element in a string array is a null string and each element in a numeric array has a value of zero.

Naming Arrays

The rules for naming array variables follow the same pattern as the rules for other type variables, namely if a variable name ends in variable type descriptor defines the variable type.

NOTE: If a DEFSTR statement is executed then a string array name need not end in a $.

Array variable names ending in % refer to integer variables.

Type/declaration tags, such as $, %, take precedence over DEFvartype all declarations.

The following statements will remove arrays from memory:

NEW, OLD, MERGE, RUN (without continue)

CALL MEMSET --sets all elements of an array to a defined value. (See command MEMSET)

Examples
100 DIM X$(30)
Reserves space in the computer's memory for 31 string numbers of the array called X$.

100 DIM D(100),B(10,9)
Reserves space in the computer's memory for 101 members of the array called D and 110 (11 times 10) members of the array called B.