

ASC

The function ASC (A\$) returns an integer giving the ASCII value of the first character of A\$. An FC error occurs (and ASC returns 0) if A\$ is the null string. For example,

```
PRINT ASC ("CAT");
67
OK
PRINT ASC ("DOG")
68
OK
```

Relations

The relations <, =, >, <=, >= and <> may be used to compare two strings as well as to compare numeric values. Two strings are equal if they are the same, that is if both have the same length and the ASCII values of corresponding characters are equal. For example, "DOG" = "DOG" is true but "DOG" = "DOG " is false; since the space is not ignored, the lengths of "DOG" and "DOG " are not the same.

A\$ < B\$ if the ASCII value of the first character of A\$ is less than that of the first character of B\$; for example, "BAT" < "C". If the first character (or characters) is the same, A\$ < B\$ if the first nonmatching character is less; for example, "DOGFOOD" < "DOGHOUSE" because "F" < "H". The null string is less than any nonnull string, so "DOG" < "DOGFOOD" (because the null string "" < "FOOD"). Since the relation < defines the usual alphabetic order on strings, you can use it to alphabetize a set of strings. A sample program which performs a bubble sort on names is given in the following section on string arrays.

The remaining relations >, <=, >= and <> are defined in the usual way in terms of = and <. For example, A\$ <> B\$ is true if A\$ = B\$ is false.

String Arrays

Extended XYBASIC lets you DIMension arrays of string variables in precisely the same way you DIMension arrays of numeric variables. If you are writing a program which processes payroll information about employees, you might use the command

```
DIM NAME$ (50)
```

at the start of the program to allocate 51 string variables called NAME\$ (0), NAME\$ (1), ... NAME\$ (50). Then these variables can be used in the program just like simple string variables. The following program gets names from the console and sorts them into alphabetical order with a simple bubble sort. A bubble sort works by interchanging pairs of values, letting the least ("lightest") values "bubble" to the top. In this program S(1) is compared to S(2), ..., S(N) to assure S(1) is the least; then S(2) is compared to S(3), ..., S(N), and so on. Notice that the DEF STR S command in line 10 allows subsequent commands to refer to the string array as S rather than S\$.