If the # signs in the stringexpression are preceded by **, any leading spaces in the number are filled with *'s:

```
100 FOR T%=1% TO 4%
110    READ A
120    PRINT USING "**##.##",A
130 NEXT T%
140 DATA 1.777,21.772,310.1,1107.666
RUN
***1.78
**21.77
*310.10
1107.67
```

If the # signs in the stringexpression are preceded by **$, a $ is placed immediately before the number and any leading spaces are filled with *'s:

```
100 FOR T%= 1% TO 4%
110    READ A
120    PRINT USING "**$#.##",A
130 NEXT T%
140 DATA 1.777,21.772,.076,310.1
RUN
**$1.78
*$21.77
***$.08
$310.10
```

If a minus sign follows the last # sign in the stringexpression, a trailing minus sign is printed for negative numbers. If a plus sign follows the last # sign in the stringexpression, a trailing plus or minus sign is printed, depending on the sign of the number. If a plus sign precedes the first # sign in the stringexpression, a plus or minus sign precedes the number when it is printed. Note that you cannot use the last option if you are using *'s or $'s:

```
100 FOR T%=1% TO 3%
110    READ A
120     PRINT USING "**#.##-  $$#.##+  +###.##",A,A,A
130 NEXT T%
140 DATA -1.777,21.772,-.076
RUN
**1.78-     $1.78-      -1.78
*21.77     $21.77+     +21.77
***.08-      $.08-       -.08
```

PRINT USING can also work with strings. If the stringexpression equals !, the first character of the string is printed:

```
100 PRINT USING "!","ABC"
RUN
A
```

If the stringexpression is a backslash followed by a backslash, two or more characters from the string are printed. The number printed equals two plus the number of spaces between the backslashes. If the string is shorter than the number of characters to be printed, spaces will be added