

RUN

Sentence? The world is everything which is the case.

The

world

is

everything

which

is

the

case.

OK

Another form of INSTR lets you specify an offset, so you can look for occurrences of a substring starting at any character of a string. INSTR (I, A\$, B\$) returns the least integer n  $\geq$  I such that the substring of A\$ starting at the nth character matches B\$. For example:

```
PRINT INSTR (4, "VOOD00", "00")
```

5

OK

The string "VOOD00" contains the substring "00" starting at positions 2 and 5, so here INSTR returns the first position greater than 4. The sample program above may be simplified by making the following changes, using this form of INSTR and leaving the value of A\$ unchanged.

```
30 LAST = I + 1
```

```
35 I = INSTR (LAST, A$, " ")
```

```
60 S$ (N) = MID$ (A$, LAST, I-LAST)
```

```
70
```

```
90 S$ (N) = MID$ (A$, LAST)
```

RUN

Sentence? There is much here to excite admiration and perplexity.

There

is

much

here

to

excite

admiration

and

perplexity.

OK

### GET\$

The GET\$ function lets you check whether a character has been typed while a program is running. If a character has been typed, GET\$ returns a string value consisting of the typed character. If no character has been typed, GET\$ returns the null string.

You can use GET\$ to let a user respond to a question by typing Y or N, or to define control characters to monitor program execution without using <control-C> and CONT. The following example just increments the value in