

Section 13: Errors

Even experienced programmers make mistakes or ask XYBASIC to do something it cannot; for example, the result of a multiplication might be too large. When XYBASIC finds an error in your program, it prints an error message giving the type and location of the error. Suppose your program contains the line

```
10 LET 3 = J
```

Since the left hand side of a LET command must be a variable, XYBASIC gives you an error message:

```
SN ERROR: 10 LET
           3 = J
```

OK

Here SN is a code indicating a SyNtax error; the two-letter codes for other errors are given below. The line in which the error occurs is then printed, with a linefeed indicating the approximate location of the error. Using this information you can correct the error and try running the program again.

TRAP and UNTRAP

Sometimes you want your program to continue even after XYBASIC detects an error. For example, you might be running an important experiment and want your program to avoid stopping. This is done by using UNTRAP mode, in which XYBASIC attempts to continue execution after reporting errors to the console. Certain errors are always fatal, i.e. there is no way to recover from them, but for other errors XYBASIC applies a particular recovery procedure and continues. To get into UNTRAP mode you just type UNTRAP, and to get out you type TRAP. XYBASIC is initially in TRAP mode, and returns to TRAP mode whenever a NEW is executed.

Try the following example.

```
NEW
20 FOR I=1 TO 3
30 MUMBLE
40 NEXT I
RUN
```

```
SN ERROR: 30 MUMBLE
```

OK

Note that XYBASIC returns to direct mode when it finds the syntax error. Now say