```
[40 N = N + 2]                N= 11
[50 FOR I = 3 TO N/2 STEP 2]                  I= 3
[60 IF N MOD I = 0 THEN 40]
[70 NEXT I]    I= 5  ^C
BREAK AT LINE 70
OK
```

After execution of the line 10 TRACE command, XYBASIC prints the bracketed line number and contents of each command before it is executed. If the command is LET, FOR, NEXT, READ or INPUT, the name and new value of the modified variable is also printed.

If a program is not working correctly, you can interrupt it with <control-C>, type TRACE and then CONTinue execution. By following the TRACE you can often find why your program does not work in the way you expected.

Execution of the UNTRACE command disables the TRACE feature, so you can TRACE precisely the sections of program you desire. Try changing the program:

```
45 UNTRACE
75 TRACE
RUN
[20 PRINT 2; "IS PRIME"]      2 IS PRIME

[30 N = 1]     N= 1
[40 N = N + 2]                N= 3
[45 UNTRACE]
[80 PRINT N; "IS PRIME"]      3 IS PRIME

[90 GOTO 40]
[40 N = N + 2]                N= 5
[45 UNTRACE]
[80 PRINT N; "IS PRIME"]      5 IS PRIME

[90 GOTO 40]
[40 N = N + 2]                N= 7
[45 UNTRACE]   ^C
BREAK AT LINE 45
OK
```

XYBASIC also disables TRACE whenever you execute a NEW.

If your system supports a LST device such as a lineprinter, you may find it convenient to use <control-P> and <control-O> to print TRACEs on the printer rather than on the console.

BREAK and UNBREAK

For some purposes TRACE provides more information than you need; some bugs are easier to find if you just check what happens when certain lines are executed or certain variables are changed. The BREAK command gives you a flexible and powerful tool to do so by letting you set breakpoints on variables or line numbers. BREAK will greatly aid you in debugging your program and let you get it running correctly much faster, increasing your