```
[50 I = I + L]
[50 I = I + L]
[50 I = I + L]
[50 I = I + L]
[50 I = I + L]
[50 I = I + L]
[50 I = I + L]
OK
```

With line breakpoints you have the option of making the break occur every few times the line is executed, rather than every time. For example:

```
10 BREAK 50, 3
RUN
[50 I = I + L]
[50 I = I + L]
[50 I = I + L]
OK
```

The break message is now printed every third time the line is executed. When setting a line break you may also specify a variable or list of variables, preceded by a semicolon. The values of the variables in the list are then printed whenever the break occurs.

```
10 BREAK 50; I
RUN
[50 I = I + L]                I= 1
[50 I = I + L]                I= 1
[50 I = I + L]                I= 2
[50 I = I + L]                I= 3
[50 I = I + L]                I= 5
[50 I = I + L]                I= 8
[50 I = I + L]                I= 13
[50 I = I + L]                I= 21
[50 I = I + L]                I= 34
[50 I = I + L]                I= 55
OK
10 BREAK 50, 4; I, J
RUN
[50 I = I + L]                I= 3  J= 4
[50 I = I + L]                I= 21  J= 8
OK
```

Finally, you can use the $ option to set a line break which returns you to direct mode (rather than continuing execution) when the line break occurs.

```
10 BREAK 50; $
RUN

BREAK AT LINE 50
OK
```

The $ option can be used in conjunction with the other BREAK options.