

```

NEW
OK
10 FOR I = 1 TO 5
20 FOR J = I TO 5
30 PRINT I, J, I*J
40 NEXT J
50 NEXT I

```

```

RUN
1      1      1
1      2      2
1      3      3
1      4      4
1      5      5
2      2      4
2      3      6
2      4      8
2      5     10
3      3      9
3      4     12
3      5     15
4      4     16
4      5     20
5      5     25
OK

```

Because this construction is so common, XYBASIC lets you combine successive NEXT commands into one. Here you can replace lines 40 and 50 with

```

40 NEXT J, I
50

```

A single NEXT can specify as many variables as there are corresponding FOR commands. You must be careful, however, to specify the correct order of nesting (NEXT J, I rather than NEXT I, J in the example), or a NF error will occur when XYBASIC encounters the wrong variable name.

If the conditions of a FOR command are initially false (namely, the initial value is greater than the bound when the increment is positive, or less than the bound when the increment is negative), the body of the FOR loop is not executed. Instead, XYBASIC searches through the program for the matching NEXT and executes the following command. For example:

```

NEW
OK
10 FOR I = 1 TO 0
20 PRINT "THIS NEVER GETS PRINTED"
30 NEXT I
40 PRINT "END OF LOOP"
RUN
END OF LOOP
OK

```

If XYBASIC is unable to find the matching NEXT in this case, a FR (For) error will occur.

In Extended XYBASIC you can write FOR loops which use either integer variables or floating point variables to control the loop. Since integer