

```

NEW
OK
10 PRINT "START OF BUTTON INSTRUCTIONS, HOPE YOU'RE READY!"
20 GOSUB 200
30 READ X
40 PRINT "PUSH THE RED BUTTON"; X; "TIMES"
50 GOSUB 200
60 READ Y
70 PRINT "PUSH THE YELLOW BUTTON"; Y; "TIMES"
80 GOSUB 200
90 I = I + 1
100 IF I < 3 THEN 30 'LOOP 3 TIMES
110 PRINT "ALL DONE";
120 END
150 DATA 2,7,13,1,10,21
200 'SUBROUTINE TO PROMPT USER
210 INPUT "TYPE 1 TO CONTINUE" N
220 IF N <> 1 THEN 210
230 RETURN
RUN
START OF BUTTON INSTRUCTIONS, HOPE YOU'RE READY!
TYPE 1 TO CONTINUE? 1
PUSH THE RED BUTTON 2 TIMES
TYPE 1 TO CONTINUE? 1
PUSH THE YELLOW BUTTON 7 TIMES
TYPE 1 TO CONTINUE? 1
PUSH THE RED BUTTON 13 TIMES
TYPE 1 TO CONTINUE? 1
PUSH THE YELLOW BUTTON 1 TIMES
TYPE 1 TO CONTINUE? 1
PUSH THE RED BUTTON 10 TIMES
TYPE 1 TO CONTINUE? 1
PUSH THE YELLOW BUTTON 21 TIMES
TYPE 1 TO CONTINUE? 1
ALL DONE
OK

```

This program reads the button pushing information from the DATA statement of line 150. When lines 30 and 60 are executed they READ the next piece of DATA (first 2 to X, then 7 to Y, then 13 to X, etc.). The counter I prevents an OD error.

### FOR and NEXT

The FOR and NEXT commands allow you to execute a group of commands several times, that is to write controlled loops. The FOR and NEXT construction greatly simplifies XYBASIC programming by doing some of the programmer's work automatically, making programs both easier to write and easier to understand. To see how FOR and NEXT work, first look at the following program using IF / THEN and GOTO.