

```
IF X = 0 AND (Y = 1 OR Z <= 10) THEN GOSUB 100
```

allows you to test for several cases with a single IF command. Second, they perform bit manipulation. For example, the Extended XYBASIC command

```
LET I% = (J% AND #F000) OR #FFF
```

sets the most significant four bits of I% to the corresponding bits of J%, and sets the other twelve bits of I% to 1. Since Integer XYBASIC only allows integer variables, this command corresponds to the Integer XYBASIC command

```
LET I = (J AND #F000) OR #FFF
```

The following program demonstrates the logical operators by printing the representations of two values and of the results of applying logical operators to them.

```
NEW
OK
10 INPUT "TYPE TWO NUMBERS" A, B
20 PRINT A, " IS", : TEMP = A : GOSUB 200
30 PRINT B, " IS", : TEMP = B : GOSUB 200
40 PRINT A; "AND"; B; "IS", : TEMP = A AND B : GOSUB 200
50 PRINT A; "OR"; B; "IS", : TEMP = A OR B : GOSUB 200
60 PRINT A; "XOR"; B; "IS", : TEMP = A XOR B : GOSUB 200
70 PRINT "NOT "; A; " IS", : TEMP = NOT A : GOSUB 200
80 GOTO 10
200 REM SUBROUTINE TO PERFORM BINARY CONVERSIONS
210 FOR I = 15 TO 0 STEP -1
220 PRINT TEST(TEMP,I);
230 NEXT I
240 PRINT
250 RETURN
RUN
TYPE TWO NUMBERS? 15,255
15 IS 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1
255 IS 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1
15 AND 255 IS 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1
15 OR 255 IS 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1
15 XOR 255 IS 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 0 0
NOT 15 IS 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 0 0
TYPE TWO NUMBERS? -1,2
-1 IS 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
2 IS 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
-1 AND 2 IS 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
-1 OR 2 IS 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
-1 XOR 2 IS 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 1
NOT -1 IS 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
TYPE TWO NUMBERS? #FFE0,65
-32 IS 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 0 0 0
65 IS 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0
-32 AND 65 IS 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0
-32 OR 65 IS 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 0 0 0
-32 XOR 65 IS 1 1 1 1 1 1 1 1 1 1 1 1 0 1 0 0 0 0
NOT -32 IS 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1
```