

it can be.

```

NEW
OK
10 PRINT "BINARY CONVERSIONS"
20 INPUT "NUMBER TO CONVERT" N
30 FOR I = 15 TO 0 STEP -1
40 PRINT TEST(N,I);
50 NEXT I
60 PRINT
70 GOTO 20
RUN
BINARY CONVERSIONS
NUMBER TO CONVERT? 45
 0 0 0 0 0 0 0 0 0 0 1 0 1 1 0 1
NUMBER TO CONVERT? #FF
 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1
NUMBER TO CONVERT? -1
 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
NUMBER TO CONVERT? 1025
 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 1
NUMBER TO CONVERT? ^C
BREAK AT LINE 20
OK

```

This program uses TEST to find the binary representation of the number you type in. Since XYBASIC accepts hexadecimal constants as well as decimal, this program converts either decimal or hexadecimal to binary, as the examples show. You can also convert numbers to binary by using the function BIN\$, described below.

Since representations of integer values have 16 bits in XYBASIC, the second argument of TEST is evaluated mod 16.

### Logical Operators

XYBASIC lets you use the logical operators AND, OR (inclusive or), XOR (exclusive or) and NOT. Each operates bitwise on its 16-bit integer arguments. That is, the value of bit *i* of A AND B will be the value of bit *i* of A ANDed with the value of bit *i* of B, where *i* = 0, 1, ..., 15. The truth table for each bit is:

A	B	A AND B	A OR B	A XOR B	NOT A
0	0	0	0	0	1
0	1	0	1	1	1
1	0	0	1	1	0
1	1	1	1	0	0

If you think of 1 as representing true and 0 as representing false, you can see that A AND B is true when both A and B are true; A OR B is true when either is true; A XOR B is true when exactly one is true; and NOT A is true when A is false.

You may want to use logical operators for two different purposes. First, they let you build complicated conditions in logical formulas. For example,