Now check that you typed everything correctly by typing

```
LIST
```

XYBASIC will then LIST your program, followed by an OK prompt. If you made a mistake, retype the bad line and it will be replaced. Now you can run your program by typing

```
RUN
```

Your program will ask you for a number by typing

```
NUMBER TO CONVERT?
```

You type 5 followed by <carriage return>, and your program will say

```
 0  0  0  0  0  0  0  0  0  0  0  0  0  1  0  1
NUMBER TO CONVERT?
```

Type some more values. When you want to stop, hold down the control key (sometimes labelled CNTL or CTRL or CTL) and type C at the same time. XYBASIC will say

```
^C
BREAK IN LINE 10
OK
```

Congratulations -- you have just run a program!

You probably noticed that every line of your program has a line number, and that line numbers are in ascending order (the lowest are first and the highest last). XYBASIC orders lines automatically, so line 10 will be LISTed before line 20 even if you type line 20 before line 10. You also may have noticed that the program has gaps between line numbers. The lines could have been numbered 1, 2, 3, 4, 5 and 6, but then if you decided to add another line between lines 2 and 3 you would have to retype and renumber lines 3, 4, 5 and 6, rather than just adding line 25.

When you typed NEW, LIST and RUN you were giving XYBASIC commands in direct mode. Direct commands have no line numbers, and XYBASIC executes them as soon as the <carriage return> is typed. When you typed in the program, you were using program mode. Program lines have line numbers, and XYBASIC adds them to the current program but does not execute them until you type RUN.

Now let's look at your first program more closely to see how it works. After you type RUN, XYBASIC executes the numbered commands one at a time in the order they appear, unless a command tells it to do otherwise. Line 10 is an INPUT command, which gets information from your terminal. It gives the variable named N the value you type. XYBASIC may change the value of a variable, but its name always remains the same. Lines 20 and 40 create a FOR loop, causing the commands in between (namely line 30) to be executed repeatedly. Line 30 uses the PRINT command to print information on your console. The value it prints is computed by TEST, a function found only in XYBASIC which enables you to look at the value of a single bit of a variable. Line 60 is a GOTO command, telling XYBASIC to go back and execute line 10 again instead of executing the following command.