Number Bases

We always use base 10 in our everyday arithmetic. However, in computer science, we tend to be a little creative in the use of foreign bases. You have probably already seen numbers written in base 2 (binary), base 8 (octal) and base 16 (hexadecimal). Let's experiment with number conversions, function calls and loops.

To illustrate, here is what the number 30 looks like in various bases from 2 through 16:

Base	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Representation	11110	1010	132	110	50	42	36	33	30	28	26	24	22	20	1e

Create a new source file, and let's have some fun!

1. Write a function `convert` that will take two integer parameters, a value and a base number. The function will return a string containing the representation of the value in this base. For example, convert(30, 2) should return "11110".

   It turns out that if you are working in Java, this function will be easy, because this functionality can already be found in the Java API.

2. In your main program, ask the user interactively to enter a value. Then, print out all of the representations of this value from base 2 through base 16, using a loop. In other words, do something like this pseudocode:

   ```
   for i = 2 to 16
      print (convert(value, i))
   ```

   Your output should be clearly labelled so that the user can easily understand it.