Problem 1 : Text book, problem 6, chapter 15, page 708
Converting decimal numbers to binary numbers and be done recursively. Write a function that takes a positive int argument and returns the corresponding binary number as an int composed of only ones and zeros. For example, **convert(5)** returns the int: **101**

The insight for this problem comes from the fact that the right most digit of a decimal n is easy to calculate. It is the remainder when dividing by the base 2: n%2. To get the next right most digit, you can take the resulting quotient, that is n_2 = n/2, and find its remainder, n_2%2, which is the next digit. Unfortunately, that generates the digits from right to left and we want them from left to right. We could easily do that non-recursively using string concatenation or string reversal, but in this exercise you are to use recursion to recursively calculate the digits. Effectively, you are letting recursion reverse the ordering: think in terms of calculating the rightmost digit as described previously, but then let recursion reverse the digits.