Here is code we claim will return, for each nonnegative integer \( n \), the equivalent binary representation.

We use \& to indicate concatenation, so for example

"Strong"\&"Induction" = "StrongInduction"

We use \( \lfloor x \rfloor \) to indicate rounding down to an integer, so that

\[
\lfloor \pi \rfloor = 3 \quad \lfloor -\pi \rfloor = -4 \quad \lfloor 12 \rfloor = 12
\]

1) function X(integer \( n \)) returns String
2) {
3)     if \( n = 0 \) return "0";
4)     else if \( n = 1 \) return "1";
5)     else{
6)         \( q = \lfloor \frac{n}{2} \rfloor \);
7)         \( r = n - 2q \);
8)         if \( r = 0 \) return X(\( q \))\&"0";
9)             else if \( r = 1 \) return X(\( q \))\&"1";
10)     }
11) }

We can prove this code is correct using strong induction.

We need to have two base cases, since \( X() \) handles both \( n = 0 \) and \( n = 1 \) as special cases. For \( n = 0 \), the condition on line (3) holds true, so the function returns the string "0", which is correct. For \( n = 1 \) the condition on line (3) is false, so line (4) executes. Since the condition there is true, the function returns "1", which is correct.

Now assume the function returns the correct value if it has any input in the range 0, 1, \ldots, \( k \), where \( k \geq 1 \). Now suppose the program is called on \( n = k + 1 \). Notice that \( n \geq 2 \) which means nothing will happen on lines (3) and (4) and execution will continue on line (6). After it assigns a value to \( q \), either

\[ 2q = n \text{ or } 2q = n - 1 \]

and in either case,

\[ 0 \leq q \leq \frac{n}{2} \leq n - 1 = k. \]

After line (7) assigns a value to \( r \), we will have

\[ n = 2q + r \]
and

\[ r = 0 \text{ or } r = 1 \]

If \( r = 0 \) the condition on line (8) is satisfied, and the function \( X \) will be called on \( q \). We know \( 0 \leq q \leq k \) and so we also know that the function returns the correct value here, i.e., \( X(q) \) will be the binary representation of \( q \). What is returned is that followed by a zero, which is the binary representation for \( 2^q \), but \( 2^q = 2^q + r = n \) so this is correct.

If \( r = 1 \) nothing happens on line (8), but the condition on line (9) will be true. We know that \( X(q) \) will return the binary representation of \( q \). What is returned is that followed by a one, which is the binary representation of \( 2^q + 1 \), and \( 2^q + 1 = 2^q + r = n \), so this is correct.