Problem 3(a): Converting numbers to binary form

\[ a := \text{convert}(654123, \text{binary}); \]
\[ a := 1001111101100101011 \]

\[ b := \text{convert}(654123, \text{base}, 2); \]
\[ b := [1, 1, 0, 1, 0, 1, 0, 1, 0, 1, 1, 0, 1, 1, 0, 0, 1] \]

The first command returns the binary expansion of 654123. (Note that this is actually an integer whose decimal expansion has only 0's and 1's.; cf. Maple's help page on "convert/binary".) The second command gives the list of binary digits (bits) of the binary expansion of 65431 in reverse order.

Problem 3(b): A program which converts a given integer \( n > 0 \) into its binary form (list of bits).

Input: an integer \( n > 0 \). Output: the list of bits of \( n \) in reverse order (as for the \text{convert}(*, \text{base}, 2) command).

\[ \text{binexp} := \text{proc}(n) \text{ local } ls, q, r; \]
\[ \text{local } ls, q, r; \]
\[ ls := [ ]; q := n; \]
\[ \text{while } (q \neq 0) \text{ do}; \]
\[ r := \text{irem}(q, 2); q := \text{iquo}(q, 2); \]
\[ ls := [\text{op}(ls), r]; \text{od}; \]
\[ \text{return } (ls); \text{ end}; \]

\[ \text{binexp} := \text{proc}(n) \]
\[ \text{local } ls, q, r; \]
\[ ls := [ ]; q := n; \]
\[ \text{while } q <> 0 \text{ do } r := \text{irem}(q, 2); q := \text{iquo}(q, 2); ls := [\text{op}(ls), r] \text{ end } \]
\[ \text{return } ls \]
\end{proc}

Testing this for \( n = 654123 \) yields:

\[ \text{binexp}(654123); \]
\[ [1, 1, 0, 1, 0, 1, 0, 1, 0, 1, 1, 0, 1, 1, 1, 0, 0, 1] \]

By inspection we see that this is the same answer as obtained by the command \text{convert}(654123, \text{base}, 2).

A better way to check this is by using MAPLE's \text{evalb} command:

\[ \text{evalb}(\text{binexp}(654123) = b); \]
\[ \text{true} \]