(machine multiplication) Given two natural numbers, write a program to find their product using only addition, subtraction, doubling, halving, test for even, and test for zero, but not multiplication or division.

§ For a solution with linear time, use the identity

\[ x \times y = (x-1) \times y + y \]

For a solution with logarithmic time, use the identities

\[ x \times y = x/2 \times y \times 2 \quad \text{(for even } x \text{)} \]
\[ x \times y = (x-1)/2 \times y \times 2 + y \quad \text{(for odd } x \text{)} \]

Let all variables be natural.

\[ x := x \times y \quad \text{if } x = 0 \text{ then } \text{ok} \]
\[ \quad \text{else if } \text{even } x \text{ then } x := x/2. \quad x := x \times y. \quad x := x \times 2. \quad x := x + y \quad \text{fi} \]
\[ \quad \text{else } x := (x-1)/2. \quad x := x \times y. \quad x := x \times 2. \quad x := x + y \quad \text{fi} \]

Note that in the solution, the occurrences of \( x := x \times y \) are recursive calls. Note also that in the usual binary representation of natural numbers, \( x := x \times 2 \) is just shift left, and both \( x := x/2 \) (for even \( x \)) and \( x := (x-1)/2 \) (for odd \( x \)) are just shift right. The execution time is \( \text{if } x = 0 \text{ then } 0 \text{ else } 1 + \text{floor} (\log x) \text{ fi} \).

Here is another solution in which the recursive calls can be implemented as branches.

Let \textit{nat} variables \( a \) and \( b \) have the given numbers as their initial values, and let \textit{nat} variable \( c \) have their product as its final value.

\[ c' = a \times b \quad \text{if } a = 0 \text{ then } \text{ok} \]
\[ \quad \text{else if } \text{even } a \text{ then } a := a/2. \quad b := b \times 2. \quad c' := c + a \times b \]
\[ \quad \text{else } c := c + b. \quad a := (a-1)/2. \quad b := b \times 2. \quad c' := c + a \times b \quad \text{fi} \quad \text{fi} \]

with execution time \( \text{if } a = 0 \text{ then } 0 \text{ else } 1 + \text{floor} (\log a) \text{ fi} \)