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
\[ xy = (x-1)y + y \]
For a solution with logarithmic time, use the identities
\[ xy = \frac{x}{2}y \times 2 \quad (\text{for even } x) \]
\[ xy = \frac{x-1}{2}y \times 2 + y \quad (\text{for odd } x) \]
Let all variables be natural.
\[ x := xy \quad \text{if } x=0 \text{ then } \text{ok} \]
\[ \text{else if even } x \text{ then } x := x/2. \quad x := xy. \quad x := x \times 2 \]
\[ \text{else } x := (x-1)/2. \quad x := xy. \quad x := x \times 2. \quad x := x + y \quad \text{fi} \]
Note that in the solution, the occurrences of \( x:= xy \) 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 \( \text{if } x=0 \text{ then } 0 \text{ else } 1 + \text{floor} (\log x) \) fi.

Here is another solution in which the recursive calls can be implemented as branches. Let \( \text{nat} \) variables \( a \) and \( b \) have the given numbers as their initial values, and let \( \text{nat} \) variable \( c \) have their product as its final value.
\[ c' = ab \quad \text{if } a=0 \text{ then } \text{ok} \]
\[ \text{else if even } a \text{ then } a := a/2. \quad b := b \times 2. \quad c' = c + ab \]
\[ \text{else } c := c+b. \quad a := (a-1)/2. \quad b := b \times 2. \quad c' = c + ab \quad \text{fi} \]
with execution time \( \text{if } a=0 \text{ then } 0 \text{ else } 1 + \text{floor} (\log a) \) fi.