(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

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

For a solution with logarithmic time, use the identities

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

Let all variables be natural.

\[
\begin{align*}
    x &:= xy & \text{if } x=0 \text{ then } \text{ok} \\
    \text{else if even } x &\text{ then } x := x/2. \; x := xy. \; x := x \times 2 \\
    \text{else } x &:= (x-1)/2. \; x := xy. \; x := x \times 2. \; x := x+y \; \text{fi fi}
\end{align*}
\]

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 \( \text{if } x=0 \text{ then } 0 \text{ else } 1 + \text{floor} \ (\log x) \; \text{fi} \).

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.

\[
\begin{align*}
    c' &:= a \times b \; \text{if } a=0 \text{ then } \text{ok} \\
    \text{else if even } a &\text{ then } a := a/2. \; b := b \times 2. \; c' := c + a \times b \\
    \text{else } c &:= c+b. \; a := a-1. \; c' := c + a \times b \; \text{fi fi}
\end{align*}
\]

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

Both of these solutions can be improved by testing for evenness before testing for zeroness. If \( a \) is not even, then it's not zero, and we save a test each iteration. Here's the second program with this improvement.

\[
\begin{align*}
    c' &:= a \times b \; \text{if } a=0 \text{ then } \text{ok} \\
    \text{else if even } a &\text{ then } a := a/2. \; b := b \times 2. \; c' := c + a \times b \\
    \text{else } c &:= c+b. \; a := a-1. \; c' := c + a \times b \; \text{fi fi}
\end{align*}
\]