A program to find whether the number of ones in the binary representation of a given natural number is even or odd.

Let the given natural number be the initial value of natural variable \( n \), and report the answer as the final value of binary variable \( p \). Define

\[
R \equiv p' = \text{even} (\Sigma i: \text{nat} \cdot \text{mod} (div n 2^i) 2) \\
Q \equiv p' = (p = \text{even} (\Sigma i: \text{nat} \cdot \text{mod} (div n 2^i) 2))
\]

Then the refinements are

\[
R \iff p := T. \ Q \\
Q \iff \text{if } n=0 \text{ then ok else } p := p = \text{even} n. \ n := \text{div} n 2. \ Q \ \text{fi}
\]

The proof of the first refinement is one use of the Substitution Law. The last refinement, last case, is

\[
\text{off}
\]

because chopping off \( i \) bits from the right end of a binary number followed by chopping off \( j \) more bits is the same as chopping off \( i+j \) bits.

The last refinement, last case, is

\[
p' = (p = \text{even} (\Sigma i: \text{nat} \cdot \text{mod} (div n 2^i) 2)) \iff n=0 \land ok \text{ \ expand } Q , \text{ two substitutions}
\]

\[
p' = (p = \text{even} (\Sigma i: \text{nat} \cdot \text{mod} (div n 2^i) 2)) \iff n>0 \land (p := p = \text{even} n. \ n := \text{div} n 2. \ Q) \text{ \ expand } Q , \text{ two substitutions}
\]

\[
p' = (p = \text{even} (\Sigma i: \text{nat} \cdot \text{mod} (div n 2^i) 2)) \iff n>0 \land p' = ((p = \text{even} n) = \text{even} (\Sigma i: \text{nat} \cdot \text{mod} (div n 2^i) 2)) \text{ \ use the piece of arithmetic; also, drop } n>0 \text{ \ (we won't need it)}
\]

\[
p' = (p = \text{even} (\Sigma i: \text{nat} \cdot \text{mod} (div n 2^i) 2)) \iff p' = ((p = \text{even} n) = \text{even} (\Sigma i: \text{nat} \cdot \text{mod} (div n 2^{i+1} 2)) \text{ \ binary = is associative}
\]

\[
(p' = p) = \text{even} (\Sigma i: \text{nat} \cdot \text{mod} (div n 2^i) 2) \iff (p' = p) = (\text{even} n = \text{even} (\Sigma i: \text{nat} \cdot \text{mod} (div n 2^{i+1} 2))) \text{ \ transparency}
\]

\[
\text{even} (\Sigma i: \text{nat} \cdot \text{mod} (div n 2^i) 2) = (\text{even} n = \text{even} (\Sigma i: \text{nat} \cdot \text{mod} (div n 2^{i+1} 2))) \text{ \ binary = is associative and symmetric}
\]

\[
\text{even} n = (\text{even} (\Sigma i: \text{nat} \cdot \text{mod} (div n 2^i) 2) = \text{even} (\Sigma i: \text{nat} \cdot \text{mod} (div n 2^{i+1} 2))) \text{ \ in the first sum, separate out } i=0
\]

\[
\text{even} n = (\text{even} (\text{mod} n 2 + \Sigma i: \text{nat} \cdot \text{mod} (div n 2^{i+1} 2) = \text{even} (\Sigma i: \text{nat} \cdot \text{mod} (div n 2^{i+1} 2))
\]

If \( n \) is even, \( \text{mod} n 2 = 0 \). If \( n \) is odd, \( \text{mod} n 2 = 1 \), changing the evenness of the upper sum.

\[
T \iff \text{if } n=0 \text{ then } t'=t \text{ else } t' \leq t + \log n \ \text{fi}
\]

Then the refinements are

\[
T \iff p := T. \ T \\
T \iff \text{if } n=0 \text{ then ok else } p := p = \text{even} n. \ n := \text{div} n 2. \ t := t+1. \ T \ \text{fi}
\]

The proof of the first refinement is one trivial use of the Substitution Law. The second refinement is proven by cases. The first case is:

\[
T \iff n=0 \land ok \text{ \ expand } T \text{ and } ok \\
\text{if } n=0 \text{ then } t'=t \text{ else } t' \leq t + \log n \ \text{fi} \iff n=0 \land n'=n \land p'=p \land t'=t \text{ \ context}
\]
\[
\begin{align*}
\text{if } 0=0 \text{ then } t' &= t \text{ else } t' \leq t + \log 0 \text{ fi} \\ &\iff n=0 \land n'=n \land p'=p \land t'=t \quad \text{simplify} \\
T &\iff n=0 \land n'=n \land p'=p \land t'=t \\
T &\quad \text{base} \\
\text{The other case is} \\
T &\iff n>0 \land (p := p = \text{even } n. \ n := \text{div } n \ 2. \ t := t+1. \ T) \quad \text{expand } T \text{ and substitute} \\
&\quad \text{if } n=0 \text{ then } t'=t \text{ else } t' \leq t + \log n \text{ fi} \\
&\iff n>0 \land \text{if } \text{div } n \ 2 = 0 \text{ then } t'=t+1 \text{ else } t' \leq t+1 + \log (\text{div } n \ 2) \text{ fi} \\
&\quad \text{use } n>0 \text{ as context} \\
\iff t' \leq t + \log n \iff n>0 \land \text{if } n=1 \text{ then } t'=t+1 \text{ else } t' \leq t+1 + \log (\text{div } n \ 2) \text{ fi} \\
&\quad \text{increase } \text{div } n \ 2 \text{ to } n/2 \\
\iff t' \leq t + \log n \iff n>0 \land \text{if } n=1 \text{ then } t'=t+1 \text{ else } t' \leq t+1 + \log (n/2) \text{ fi} \\
&\quad \text{use context } n=1 \text{ in } \text{then } \text{ part, and } \log \text{ law in } \text{else } \text{ part} \\
\iff t' \leq t + \log n \iff n>0 \land \text{if } n=1 \text{ then } t'=t+\log n \text{ else } t' \leq t + \log n \text{ fi} \\
&\quad \text{case idempotent and specialize} \\
\iff T \\
\end{align*}
\]