245 (parity check) Write a program to find whether the number of ones in the binary representation of a given natural number is even or odd.

After trying the question, scroll down to the solution.

§ 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

$R = p' = even\ (\Sigma i\colon nat\cdot mod\ (div\ n\ 2^i)\ 2)$

$Q = p' = (p = even\ (\Sigma i\colon nat\cdot mod\ (div\ n\ 2^i)\ 2))$

Then the refinements are

$R \Longleftarrow p{:=}\top.\ Q$

$Q \Longleftarrow$ **if** $n{=}0$ **then** ok **else** $p{:=}\ p = even\ n.\ \ n{:=}\ div\ n\ 2.\ \ Q$ **fi**

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

The last refinement can be proven by cases. The first case is

$p' = (p = even\ (\Sigma i\colon nat\cdot mod\ (div\ n\ 2^i)\ 2)) \ \Leftarrow\ n{=}0 \wedge ok$ — expand ok

$=\ p' = (p = even\ (\Sigma i\colon nat\cdot mod\ (div\ n\ 2^i)\ 2)) \ \Leftarrow\ n{=}0 \wedge p'{=}p \wedge n'{=}n$ — context

$=\ p = (p = even\ (\Sigma i\colon nat\cdot mod\ (div\ 0\ 2^i)\ 2)) \ \Leftarrow\ n{=}0 \wedge p'{=}p \wedge n'{=}n$ — simplify

$=\ p = (p = \top) \ \Leftarrow\ n{=}0 \wedge p'{=}p \wedge n'{=}n$ — identity

$=\ p = p \ \Leftarrow\ n{=}0 \wedge p'{=}p \wedge n'{=}n$ — $=$ is reflexive

$=\ \top \ \Leftarrow\ n{=}0 \wedge p'{=}p \wedge n'{=}n$ — base

$=\ \top$

Just before doing the last case, here is a piece of arithmetic.

$div\ (div\ n\ 2^i)\ 2^j \ = \ div\ n\ 2^{i+j}$

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

$p' = (p = even\ (\Sigma i\colon nat\cdot mod\ (div\ n\ 2^i)\ 2))$

$\Leftarrow\ n{>}0 \wedge (p{:=}\ p = even\ n.\ \ n{:=}\ div\ n\ 2.\ \ Q)$ — expand Q , two substitutions

$=\ p' = (p = even\ (\Sigma i\colon nat\cdot mod\ (div\ n\ 2^i)\ 2))$

$\Leftarrow\ n{>}0 \wedge p' = ((p = even\ n) = even\ (\Sigma i\colon nat\cdot mod\ (div\ (div\ n\ 2)\ 2^i)\ 2))$

use the piece of arithmetic; also, drop $n{>}0$ (we won't need it)

$\Longleftarrow\ p' = (p = even\ (\Sigma i\colon nat\cdot mod\ (div\ n\ 2^i)\ 2))$

$\Leftarrow\ p' = ((p = even\ n) = even\ (\Sigma i\colon nat\cdot mod\ (div\ n\ 2^{i+1})\ 2))$

binary $=$ is associative

$=\ (p'{=}p) = even\ (\Sigma i\colon nat\cdot mod\ (div\ n\ 2^i)\ 2)$

$\Leftarrow\ (p'{=}p) = (even\ n = even\ (\Sigma i\colon nat\cdot mod\ (div\ n\ 2^{i+1})\ 2))$ — transparency

$=\ even\ (\Sigma i\colon nat\cdot mod\ (div\ n\ (2^i))\ 2) = (even\ n = even\ (\Sigma i\colon nat\cdot mod\ (div\ n\ 2^{i+1})\ 2))$

binary $=$ is associative and symmetric

$=\ even\ n = (even\ (\Sigma i\colon nat\cdot mod\ (div\ n\ 2^i)\ 2) = even\ (\Sigma i\colon nat\cdot mod\ (div\ n\ 2^{i+1})\ 2))$

in the first sum, separate out $i{=}0$

$=\ even\ n \ = \ (\ \ even\ (mod\ n\ 2\ +\ \Sigma i\colon nat\cdot mod\ (div\ n\ 2^{i+1})\ 2$

$\qquad = \ even\ (\Sigma i\colon nat\cdot mod\ (div\ n\ 2^{i+1})\ 2)\qquad)$

If n is even, $mod\ n\ 2\ =\ 0$. If n is odd, $mod\ n\ 2\ =\ 1$, changing the evenness of the upper sum.

$=\ \top$

Now for the timing. Define

$T =$ **if** $n{=}0$ **then** $t'{=}t$ **else** $t' \le t + log\ n$ **fi**

Then the refinements are

$T \Longleftarrow p{:=}\top.\ T$

$T \Longleftarrow$ **if** $n{=}0$ **then** ok **else** $p{:=}\ p = even\ n.\ \ n{:=}\ div\ n\ 2.\ \ t{:=}\ t{+}1.\ \ T$ **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:

$T \ \Leftarrow\ n{=}0 \wedge ok$ — expand T and ok

$=$ **if** $n{=}0$ **then** $t'{=}t$ **else** $t' \le t + log\ n$ **fi** $\ \Leftarrow\ n{=}0 \wedge n'{=}n \wedge p'{=}p \wedge t'{=}t$ — context

$=$ **if** $0{=}0$ **then** $t{=}t$ **else** $t \le t + log\ 0$ **fi** $\ \Leftarrow\ n{=}0 \wedge n'{=}n \wedge p'{=}p \wedge t'{=}t$ — simplify

$=\ \top \ \Leftarrow\ n{=}0 \wedge n'{=}n \wedge p'{=}p \wedge t'{=}t$ — base

$=\ \top$

The other case is

$$
\begin{array}{ll}
 & T \;\Leftarrow\; n>0 \land (p:= p = even\ n.\;\; n:= div\ n\ 2.\;\; t:= t+1.\;\; T) \quad \text{expand } T \text{ and substitute} \\
= & \quad \textbf{if } n=0 \textbf{ then } t'=t \textbf{ else } t' \le t + log\ n \textbf{ fi} \\
 & \Leftarrow\; n>0 \land \textbf{if } div\ n\ 2 = 0 \textbf{ then } t'=t+1 \textbf{ else } t' \le t+1 + log\ (div\ n\ 2) \textbf{ fi} \\
 & \hfill \text{use } n>0 \text{ as context} \\
= & t' \le t + log\ n \;\Leftarrow\; n>0 \;\land\; \textbf{if } n=1 \textbf{ then } t'=t+1 \textbf{ else } t' \le t+1 + log\ (div\ n\ 2) \textbf{ fi} \\
 & \hfill \text{increase } div\ n\ 2 \text{ to } n/2 \\
\Leftarrow & t' \le t + log\ n \;\Leftarrow\; n>0 \;\land\; \textbf{if } n=1 \textbf{ then } t'=t+1 \textbf{ else } t' \le t+1 + log\ (n/2) \textbf{ fi} \\
 & \hfill \text{use context } n=1 \text{ in } \textbf{then} \text{ part, and } log \text{ law in } \textbf{else} \text{ part} \\
= & t' \le t + log\ n \;\Leftarrow\; n>0 \;\land\; \textbf{if } n=1 \textbf{ then } t'=t + log\ n \textbf{ else } t' \le t + log\ n \textbf{ fi} \\
 & \hfill \text{case idempotent and specialize} \\
= & \top
\end{array}
$$