Let \( P: \text{nat} \to \text{bin} \).

(a) Define quantifier \( \text{FIRST} \) so that \( \text{FIRST} m: \text{nat} \cdot Pm \) is the smallest natural \( m \) such that \( Pm \), and \( \infty \) if there is none.

\[
\text{FIRST} m: \text{nat} \cdot Pm = \text{MIN} m: (\langle m: \text{nat} \cdot Pm \rangle \cdot m)
\]

(b) Prove \( n := \text{FIRST} m: \text{nat} \cdot Pm \iff n := 0. \) \text{while} \( \neg Pn \) \text{do} \( n := n + 1 \) \text{od}.

I suppose \( n \) is the only variable, and I prove two refinements:

\[
n := \text{FIRST} m: \text{nat} \cdot Pm \iff n := 0. \quad n := \text{FIRST} m: \text{nat}+n \cdot Pm \quad n := \text{FIRST} m: \text{nat}+n \cdot Pm
\]

\[
\text{if} \neg Pn \text{ then } n := n + 1. \quad n := \text{FIRST} m: \text{nat}+n \cdot Pm \text{ else } \text{ok fi}
\]

Proof of first refinement is substitution law. Proof of last refinement, in two cases. First case:

\[
\neg Pn \land (n := n + 1. \quad n := \text{FIRST} m: \text{nat}+n \cdot Pm)
\]

expand assignment and substitution

\[
\neg Pn \land n' = \text{FIRST} m: \text{nat}+n+1 \cdot Pm
\]

Since \( \neg Pn \) we can increase the domain of \( \text{FIRST} \)

\[
\Rightarrow n := \text{FIRST} m: \text{nat}+n \cdot Pm
\]

Last refinement last case:

\[
Pn \land \text{ok}
\]

expand \( \text{ok} \)

\[
Pn \land n' = n
\]

\[
Pn \land n' = \text{FIRST} m: \text{nat}+n \cdot Pm
\]

use assignment form, and specialize

\[
\Rightarrow n := \text{FIRST} m: \text{nat}+n \cdot Pm
\]

Although the question didn't ask for execution time, the recursive time is

\[
i' = i + \text{FIRST} m: \text{nat} \cdot Pm
\]