Let $n$ and $d$ be nat variables. Here is a refinement.

$$n' = n + d \times (d-1)/2 \iff$$

- if $d=0$ then ok
- else $d := d-1$. $n := n + d$. $n' = n + d \times (d-1)/2$

(a) Prove it.
(b) Insert appropriate time increments according to the recursive measure, and write an appropriate timing specification and refinement.
(c) Prove the timing refinement.

After trying the question, scroll down to the solution.
(a) Prove it.
§ By cases. First case.
\[ d = 0 \land ok \Rightarrow n' = n + d \times (d-1)/2 \] expand \( ok \)
\[ = d = 0 \land n' = n \land d' = d \Rightarrow n' = n + d \times (d-1)/2 \] use \( d = 0 \) as context in consequent
\[ = d > 0 \land n' = n \land d' = d \Rightarrow n' = n + 0 \times (0-1)/2 \] arithmetic and specialize
\[ \Rightarrow \top \]
Last case.
\[ d > 0 \land (d := d - 1. \ n := n + d. \ n' = n + d \times (d-1)/2) \] substitution law twice
\[ = d > 0 \land n' = n + d - 1 + (d-1) \times (d-2)/2 \] arithmetic
\[ = d > 0 \land n' = n + d \times (d-1)/2 \] specialize
\[ \Rightarrow n' = n + d \times (d-1)/2 \]

(b) Insert appropriate time increments according to the recursive measure, and write an appropriate timing specification and refinement.
§
\[ t' = t + d \iff \text{if } d = 0 \text{ then } ok \text{ else } d := d - 1. \ n := n + d. \ t := t + 1. \ t' = t + d \text{ fi} \]

(c) Prove the timing refinement.
§ Proof by cases. First case:
\[ d = 0 \land ok \Rightarrow t' = t + d \] expand \( ok \)
\[ = d = 0 \land n' = n \land d' = d \land t' = t \Rightarrow t' = t + d \] use antecedent as context in consequent
\[ = d = 0 \land n' = n \land d' = d \land t' = t \Rightarrow t = t + 0 \] arithmetic and specialize
\[ \Rightarrow \top \]
Last case.
\[ d > 0 \land (d := d - 1. \ n := n + d. \ t := t + 1. \ t' = t + d) \] substitution law 3 times
\[ = d > 0 \land t' = t + 1 + d - 1 \] arithmetic
\[ = d > 0 \land t' = t + d \] specialize
\[ \Rightarrow t' = t + d \]