Problem 1 ($\lambda$-calculus)

In this problem, we will encode arithmetic on natural numbers in the $\lambda$-calculus. The numbers themselves will be represented as follows:

- $\lambda s.\lambda z.z$ is zero,
- $\lambda s.\lambda z.s \ z$ is one,
- $\lambda s.\lambda z.s \ (s \ z)$ is two,
- $\lambda s.\lambda z.s \ (s \ (s \ z))$ is three,

... 

1. We have seen that the successor function can be defined by $\lambda n.\lambda s.\lambda z.s \ (n \ s \ z)$.

2. Define the addition function in $\lambda$-calculus.

Problem 2 (Hoare logic)

Consider the program:

Algorithm 1  
\begin{itemize}
  \item \textbf{Input:} Two integers $M > 0$ and $N \geq 0$
  \item \textbf{Output:} An integer $k$ such that $k = M^N$
  \item $a \leftarrow M$
  \item $b \leftarrow N$
  \item $k \leftarrow 0$
  \item \textbf{while} $b > 0$ \textbf{do}
      \item \textbf{if} \text{even}(b) \textbf{then}
          \item $a \leftarrow a \times a$
          \item $b \leftarrow b/2$
      \item \textbf{else}
          \item $k \leftarrow k \times a$
          \item $b \leftarrow b - 1$
      \item \textbf{end if}
  \item \textbf{end while}
\end{itemize}

Note that $/$ is an integer division. For instance, the predicate \text{even}(b) can be implemented as $b = 2 \times (b/2)$.

Find an invariant that is strong enough to prove the correctness of the program.