1. Prove that the program segment

\[
\begin{align*}
\text{fac} & := 1 \\
\text{i} & := 2 \\
\text{while } \text{i} \leq n \text{ do} \\
\text{fac} & := \text{fac} \times \text{i} \\
\text{i} & := \text{i} + 1
\end{align*}
\]

is partially correct with respect to precondition "n \geq 1" and postcondition "fac = n!".

**Be explicit about your loop invariant:** I = "fac = (i - 1)! \land i \leq n + 1"

And this loop terminates because

\[
\begin{align*}
\text{fac} & := \text{fac} \times \text{i} \\
\text{i} & := \text{i} + 1
\end{align*}
\]

Thus since the integer values of n-i must strictly descend after each execution of the loop, eventually we must have n-i < 0 and the loop terminates.
2. Prove that the program segment

\[
i := 1 \\
\textbf{while } key_i \neq \text{test} \textbf{ do} \\
i := i+1
\]

is partially correct with respect to precondition " \((n \geq 1) \land (\exists i \ni 1 \leq i \leq n \land key_i = \text{test})" \) and postcondition " \((key_i = \text{test}) \land (1 \leq j < i \Rightarrow key_j \neq \text{test})" \).

**Be explicit about your loop invariant:** \( I = "1 \leq j < i \Rightarrow key_j \neq \text{test} " \)

\[
i := 1 \quad (n \geq 1) \land (\exists i \ni 1 \leq i \leq n \land key_i = \text{test}) \\
\textbf{while } key_i \neq \text{test} \quad 1 \leq j < i + 1 \Rightarrow key_j \neq \text{test} \\
i := i+1 \quad (key_i = \text{test}) \land (1 \leq j < i \Rightarrow key_j \neq \text{test})
\]

And this loop terminates because

\[
i := 1 \quad (n \geq 1) \land (\exists j \ni 1 \leq j \leq n \land key_j = \text{test}) \\
\textbf{while } key_i \neq \text{test} \quad \exists j \ni i \leq j \leq n \land key_j = \text{test} \\
i := i+1 \quad (\exists j \ni i + 1 \leq j \leq n \land key_j = \text{test}) \land (n-1 < n-i)
\]

The quantity \( n-i \) is strictly decreasing. Should it ever become negative then the assertion \( \exists j \ni i \leq j \leq n \land key_j = \text{test} \) would be falsified. Yet this assertion is true at the top of every pass through the loop. Therefore, there must be termination.
3. Prove the following code is partially correct with respect to precondition “$n \geq 1$” and postcondition “$(k/2 < n) \land (k \geq n) \land (\exists j \geq 0 \exists k = 2^j)$” (assume $k$ and $n$ are integer variables):

```java
k := 1
while k < n do
    k := 2*k
endwhile
```

**Be explicit about your loop invariant:** $1 = (k/2 < n) \land (\exists j \geq 0 \exists k = 2^j)$

And this loop terminates because

```java
k := 2*k
```

Thus since the integer values of $n-k$ must strictly descend after each execution of the loop, eventually we must have $n-k \leq 0$ and the loop terminates.