We practice finding a loop condition \( B \) by using the second loopy question: is \( \neg B && P \implies R \) true? Thus, we look for \( \neg B \) that makes \( \neg B && P \implies R \) true and complement \( \neg B \) to get \( B \).

Here’s the invariant \( P \) and postcondition for our first example.

\[
\begin{align*}
P: & \text{ s is the sum of } m..k-1 \text{ and } m \leq k \leq n \\
R: & \text{ s is the sum of } m..n-1
\end{align*}
\]

Knowing that \( P \) is true, and doing some pattern matching with \( P \) and \( R \), we see that \( R \) will be true if \( k = n \). Therefore, \( \neg B \) is \( k = n \), so the loop condition \( B \) is \( k \neq n \). Looking at the restriction on \( k \) in invariant \( P \), we can write the loop condition at \( k < n \) if we want. Thus, we use either

\[
\text{while } (k \neq n) \{ \ldots \} \quad \text{or} \quad \text{while } (k < n) \{ \ldots \}
\]

**A second example**

Here are the invariant and postcondition for a loop to calculate the minimum value in array segment \( b[0..n-1] \):

\[
\begin{align*}
P: & \text{ v = minimum of } b[0..k-1] \text{ and } 0 \leq k \leq n \\
R: & \text{ v = minimum of } b[0..n-1]
\end{align*}
\]

Using reasoning like we did the first example, you can see that we get the same answer for \( B \) as in the previous example.

\[
\text{while } (k \neq n) \{ \ldots \} \quad \text{or} \quad \text{while } (k < n) \{ \ldots \}
\]

**Computing \( z = b^c \)**

Here are the invariant and postcondition for a loop to store \( b^c \) in \( z \), given \( c \geq 0 \):

\[
\begin{align*}
P: & \text{ } b^c = z * x^y \text{ and } y \geq 0 \\
R: & \text{ } z = b^c
\end{align*}
\]

Again doing pattern matching, we see that \( R \) will be true when \( P \) is true and \( x^y = 1 \). That last formula, \( x^y = 1 \), is true, when \( y = 0 \). So our loop condition is \( y \neq 0 \):

\[
\text{while } (y \neq 0) \{ \ldots \}
\]

**Exercises**

In the two examples below, find the loop condition. Answers are at the end of the pdf script for this video.

1. \( P: \text{ s is the sum of } k..n-1 \text{ and } m \leq k \leq n \)  
   \( R: \text{ s is the sum of } m..n-1 \)
2. \( P: \text{ v = minimum of } b[k..n] \text{ and } 0 \leq k \leq n \)  
   \( R: \text{ v = minimum of } b[0..n] \text{ and } 0 \leq k \leq n \)

**Answers**

In the first exercise, doing pattern matching on \( P \) and \( R \), we see that \( k = m \) is needed. Therefore the loop condition is \( k = m \). This can be written as \( m < k \) if you want, since \( m \leq k \leq n \):

\[
\text{while } (k = m) \{ \ldots \} \quad \text{or} \quad \text{while } (m < k) \{ \ldots \}
\]

In the second exercise, pattern matching on \( P \) and \( R \), we see that \( k = 0 \) is needed. Therefore, the loop condition condition is \( k = 0 \). This can be written as \( 0 < k \) if you want, since \( 0 \leq k \leq n \):

\[
\text{while } (k = 0) \{ \ldots \} \quad \text{or} \quad \text{while } (m < k) \{ \ldots \}
\]